



Université Cheikh Anta DIOP de Dakar

Faculté des Sciences et Techniques

Département Mathématiques et Informatique

Laboratoire d'Algèbre de Cryptologie de
Géométrie Algébrique et Applications

LACGAA

Licence Transmission de Données et Sécurité de l'Information

Option : MCS

Thème :

HADOOP et la SÉCURITÉ

Présenté et soutenu par:
Mr Mamadou Mourtalla NDJITTE

Encadreur :
Mme Ba Aminata Ngom

Jury :

Président : Mr Mame Demba Cisse	TDSI
Membres : Mme Ba Aminata Ngom	TDSI
Mr Pape Modou Gane Ndiaye	TDSI
Mr Michel Seck	TDSI

Année Académique 2019 – 2020

Dédicace

On dédie ce travail premièrement à mes chères parents
Pour tout l'amour dont vous nous avez entouré, pour
tout ce que vous avez fait pour nous.

Que ce modeste travail, soit l'exaucement de vos vœux
tant formulés et de vos prières quotidiennes.

Que dieu, le tout puissant, vous préservé et vous procure
la santé et la longue vie afin que nous puissions à notre
tour de vous combler.

On dédie ce travail à nos chers professeurs et à l'école
TDSI.

Remerciements

En premier, nous aimerions remercier le bon Dieu le tout puissant de nous avoir donné le courage et la volonté de réaliser ce projet.

Nous désirons remercier nos chers parents qui nous ont soutenus et encouragé durant toute notre vie et pendant notre cursus d'étude.

Nos remerciements les plus chaleureux vont à **Mme Ba Aminata Ngom** pour sa disponibilité et ces très précieux conseils ainsi que ces remarques qui nous ont permis d'améliorer la qualité de ce travail.

Nous tenons à exprimer toute notre grande gratitude aux membres de jury d'avoir accepté de juger ce travail.

Nous remercions toutes les personnes ayant contribué de près ou de loin à l'élaboration de ce modeste travail.

Table des matières

I	GÉNÉRALITÉ SUR LE BIG DATA	9
1	Introduction Generale	10
1.1	BIG DATA	11
1.1.1	Définition	11
1.1.2	Big Data : l'analyse de données en masse	11
1.1.3	Les évolutions technologiques derrière le Big Data	12
1.1.4	Les principaux acteurs du marché	12
1.1.5	L'avenir du Big Data	13
1.2	HADOOP	14
1.2.1	Définition	14
1.2.2	Composant : MapReduce , HDFS ,Hadoop Common , Yarn	14
1.2.3	Avantage de Hadoop	17
1.2.4	Inconvénients de Hadoop	17
1.2.5	Hadoop et Son ecosysteme	18
1.2.6	Pourquoi la sécurité Hadoop est-elle importante?	23
1.2.7	Les trois A de la sécurité et de la protection des données	23
1.2.8	Types de sécurité Hadoop	23
2	Installation de Hadoop	24
2.1	Mode de Fonctionnement	24
2.2	Oracle Big Data Lite	24
2.2.1	Installation	25
2.2.2	Configuration	29
2.3	Hortonworks	31
2.3.1	Installation	31
2.3.2	Configuration	32
2.3.3	Access Terminal	33
2.3.4	Page d'accueil	34
2.3.5	Ambari : Définition et Réinitialisation du mot de passe administrateur . . .	34

II	Sécurisation de Hadoop	37
2.4	Autorisation de fichier	38
2.5	Access Control liste (ACL)	41
2.6	Authentification avec kerberos	42
2.7	Conclusion	50
2.8	Webographie	50

Table des figures

1.1	Big Data	11
1.2	trois v du Big Data	11
1.3	acteurs du Big Data	12
1.4	Hadoop	14
1.5	MapReduce	14
1.6	HDFS	15
1.7	Common	16
1.8	Yarn	16
1.9	Ecosysteme Hadoop	18
1.10	Spark	19
1.11	Hive	19
1.12	Pig	20
1.13	HBase	20
1.14	Sqoop	21
1.15	Storn	21
1.16	Zookeeper	22
1.17	Oozie	22
2.1	Big Data lite	25
2.2	Virtualbox	25
2.3	Telechargement Big Data lite	26
2.4	Execution Hadoop	26
2.5	Demarrage de la machine	27
2.6	Big Data lite 2	28
2.7	connection Big Data lite	28
2.8	Service Big Data lite	29
2.9	Demarrage de Cloudera	29
2.10	Connection Cloudera	30
2.11	Cloudera	30
2.12	Distribution de linux	31

2.13 Hortonworks	31
2.14 Docker + HDP	31
2.15 Accueil hortonworks	34
2.16 Ambari	35
2.17 Reset password ambari	36
2.18 POSIX	38

LEXIQUE

- HDFS = Hadoop Distributed File System
- YARN = Yet Another Resource Negotiator
- Sqoop = SQL-to-Hadoop
- ACL = Access Control liste
- AS (Authentication Service)
- TGS (Ticket Granting Service)
- Ktgs : Clé secrète du TGS connu du TGS et de l'AS.
- KsessionTGS : Clé de session entre l'utilisateur et le TGS.
- Ksession : clé secrète de session entre l'utilisateur et le service demandé.
- TicketService : Ticket d'accès au service demandé.
- TGT : Ticket d'accès au TGS.
- TS : Ticket d'accès au service demandé.
- Kutilisateur : Clé secrète de l'utilisateur, connu de l'utilisateur et du TGS.
- Kservice : Clé secrète du service demandé, connu du service et du TGS.

Première partie

GÉNÉRALITÉ SUR LE BIG DATA

Chapitre 1

Introduction Generale

Aujourd'hui, l'explosion des données est une réalité de l'univers numérique et la quantité de données augmente considérablement, même à chaque seconde. De nombreuses organisations implémentent Hadoop dans des environnements de production. Alors que les organisations se lancent dans la mise en œuvre du Big Data, la sécurité du Big Data est l'une des préoccupations majeures.

Transactions financières, banque personnelle les informations de compte et fiscales, les dossiers médicaux et autres types de données similaires sont exactement ce que les méchants recherchent. Parce que Hadoop est désormais utilisé dans le commerce de détail, la banque et la santé. applications de soins, il a également attiré l'attention des voleurs. Et si les données sont une cible juteuse, les mégadonnées peuvent être les plus importantes et les plus juteuses de toutes. Hadoop recueille plus de données à partir de plus d'endroits, les combine et les analyse de plus de façons que n'importe quel système précédent, jamais. Cela crée une valeur considérable en le faisant. Il est donc clair que la «sécurité Hadoop» est un gros problème.

Dans ce mémoire nous avons proposer un ensemble de solutions qui va assurez la sécurité de Hadoop. Pour ce faire , nous avons donc dans un premier temps fait un petit rappel sur le Big Data et la technologie Hadoop . Ensuite nous avons dans un second temps fait l'installation de Hadoop avec Big data lite et Hortonworks avant de terminer avec la mise en œuvre.

1.1 BIG DATA

1.1.1 Définition

Les mégadonnées sont un terme décrit pour une énorme quantité de données , structurées et non structurées . Les mégadonnées sont des ressources d'informations volumineuses , rapides et de différentes variétés qui nécessitent une plate-forme innovante pour des informations et une prise de décision améliorées .Les BIG DATA c'est aussi une démarche (ou un ensemble de technologies, d'architectures, d'outils et de procédures) qui consiste à collecter puis à traiter en temps réel des énormes volumes proviennent de sources, diverses structurées et non structurées, difficilement gérables avec des solutions classiques de stockage et de traitement. Hadoop est l'outil Big Data le plus populaire et le plus demandé qui résout les problèmes liés au Big Data.



FIGURE 1.1 – Big Data

1.1.2 Big Data : l'analyse de données en masse

Inventé par les géants du web, le Big Data se présente comme une solution dessinée pour permettre à tout le monde d'accéder en temps réel à des bases de données géantes. Il vise à proposer un choix aux solutions classiques de bases de données et d'analyse (plate-forme de Business Intelligence en serveur SQL...).

Selon le Gartner, ce concept regroupe une famille d'outils qui répondent à une triple problématique dite règle des 3V. Il s'agit notamment d'un **Volume** de données considérable à traiter, une grande **Variété** d'informations (venant de diverses sources, non-structurées, organisées, Open...), et un certain niveau de **Vélocité** à atteindre, autrement dit de fréquence de création, collecte et partage de ces données.

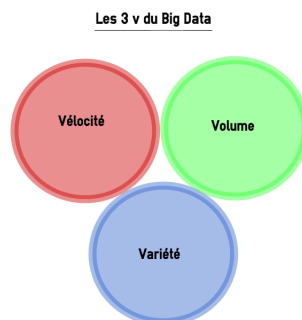


FIGURE 1.2 – trois v du Big Data

1.1.3 Les évolutions technologiques derrière le Big Data

Les créations technologiques qui ont facilité la venue et la croissance du Big Data peuvent globalement être catégorisées en deux familles : d'une part, les technologies de stockage, portées particulièrement par le déploiement du Cloud Computing. D'autre part, l'arrivée de technologies de traitement ajustées, spécialement le développement de nouvelles bases de données adaptées aux données non-structurées (Hadoop) et la mise au point de modes de calcul à haute performance (MapReduce).

Il existe plusieurs solutions qui peuvent entrer en jeu pour optimiser les temps de traitement sur des bases de données géantes à savoir les bases de données NoSQL (comme MongoDB, Cassandra ou Redis), les infrastructures du serveur pour la distribution des traitements sur les nœuds et le stockage des données en mémoire :

La première solution permet d'implémenter les systèmes de stockage considérés comme plus performants que le traditionnel SQL pour l'analyse de données en masse (orienté clé/valeur, document, colonne ou graphe).

La deuxième est aussi appelée le traitement massivement parallèle. Le Framework Hadoop en est un exemple. Celui-ci combine le système de fichiers distribué HDFS, la base NoSQL HBase et l'algorithme MapReduce.

Quant à la dernière solution, elle accélère le temps de traitement des requêtes.

1.1.4 Les principaux acteurs du marché

La filière Big Data en a attiré plusieurs. Ces derniers se sont positionnés rapidement dans divers secteurs. Dans le secteur IT, on retrouve les fournisseurs historiques de solutions IT comme Oracle, HP, SAP ou encore IBM. Il y a aussi les acteurs du Web dont Google, Facebook, ou Twitter. Quant aux spécialistes des solutions Data et Big Data, on peut citer MapR, Teradata, EMC ou Hortonworks. CapGemini, Sopra, Accenture ou Atos sont des intégrateurs, toujours des acteurs principaux dans les méga données. Dans le secteur de l'analytique, comme éditeurs BI, on peut citer SAS, Micro-strategy et Qliktech. Cette filière comporte aussi des fournisseurs spécialisés dans l'analytique comme Datameer ou Zettaset. En parallèle à ces principaux participants, de nombreuses PME spécialisées dans le Big Data sont apparues, sur toute la chaîne de valeur du secteur. En France, les pionniers ont été Hurence et Dataiku pour les équipements et logiciels de Big Data ; Criteo, Squid, Captain Dash et Tiny Clues pour l'analyse de données et Ysance pour le conseil.

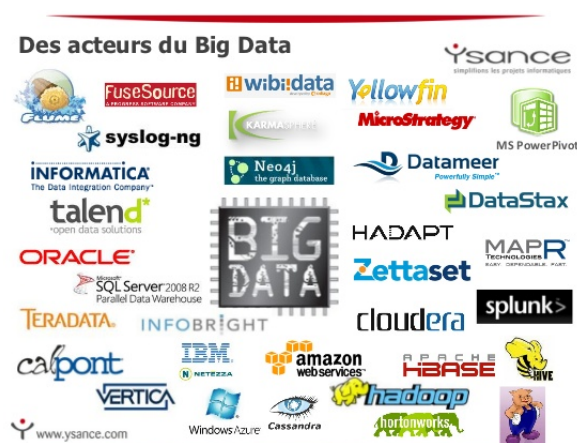


FIGURE 1.3 – acteurs du Big Data

1.1.5 L'avenir du Big Data

Etant une tendance lourde, le Big Data n'est pas une mode. Dans le domaine de l'usage, il satisfait une nécessité de travailler la donnée plus profondément, pour créer de la valeur, conjointement à des aptitudes technologiques qui n'existaient pas dans le passé. Cependant, compte tenu de l'évolution des technologies qui ne semble pas vouloir s'estomper, on ne peut pas alors parler d'une norme véritable ou de standards dans le domaine du Big data. Beaucoup d'applications du Big Data n'en sont qu'à leurs préludes et on peut s'attendre à voir apparaître des utilisations auxquelles on ne s'attend pas encore aujourd'hui. En quelque sorte, le Big Data est un tournant pour les organisations au moins aussi important qu'internet en son temps. Chaque entreprise doit donc s'y mettre dès maintenant. Dans le cas contraire, il y a un risque qu'elle se rendent comptent d'ici quelques années qu'elles se sont faites dépasser par la concurrence. Les gouvernements et les organismes publics se penchent également sur la question à travers l' open data. D'ici quelques années, le marché du big data va se mesurer en centaines de milliards de dollars. C'est un nouvel eldorado pour le business. Selon des études, il s'agit même d'une vague de fond où l'on retrouve la combinaison de la BI (business intelligence), de l'analytics et de l'internet des objets. IDC affirme qu'il devrait passer au-delà des 125 milliards de dollars avant la fin 2015. En effet, plusieurs études affluent sur cette affirmation et toutes confirment que les budgets que les entreprises vont consacrer au Big Data ne vont connaître que des fortes progressions.

1.2 HADOOP

1.2.1 Définition

Hadoop est une plate-forme informatique open source gérant le traitement et le stockage de gigantesques volumes de données pour les applications Big Data dans des clusters évolutifs de serveurs informatiques., structurées et non structurées , dans le cadre d'un système distribué .

- Données structurées - Données relationnelles.
- Données semi-structurées - données XML.
- Données non structurées - Word, PDF, texte, journaux multimédias

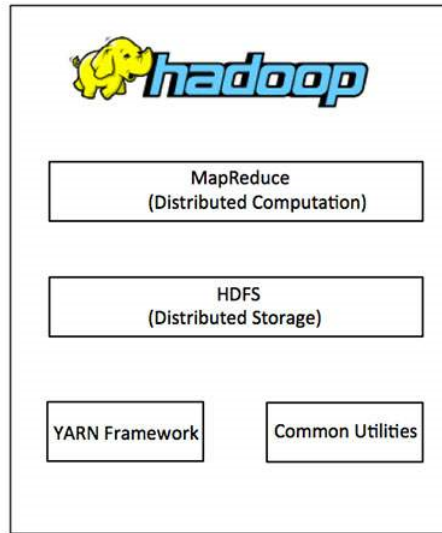


FIGURE 1.4 – Hadoop

1.2.2 Composant : MapReduce , HDFS ,Hadoop Common , Yarn

Hadoop est composée de :

1. **MapReduce** : Il s'agit d'un système basé sur YARN pour le traitement parallèle de grands ensembles de données. Le terme MapReduce fait en fait référence aux deux tâches différentes suivantes que les programmes Hadoop effectuent :
 - La tâche de mappage : il s'agit de la première tâche, qui prend les données d'entrée et les convertit en un ensemble de données, où les éléments individuels sont décomposés en tuples (paires clé / valeur).
 - La tâche de réduction : cette tâche prend la sortie d'une tâche de mappage en entrée et combine ces tuples de données en un plus petit ensemble de tuples. La tâche de réduction est toujours effectuée après la tâche de mappage

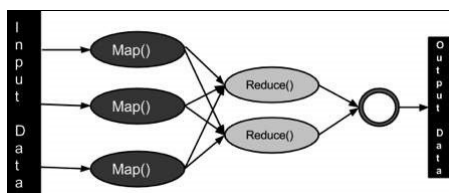


FIGURE 1.5 – MapReduce

2. **HDFS** : Un système de fichiers distribué qui fournit un accès à haut débit aux données d'application. Il fournit des autorisations de fichiers et une authentification. Les principaux composants de HDFS sont NameNode et DataNode .
- NameNode : c'est le matériel de base qui contient le système d'exploitation GNU / Linux et le logiciel namenode. Il s'agit d'un logiciel qui peut être exécuté sur du matériel standard. Le système ayant le namenode agit comme serveur maître et il effectue les tâches suivantes
 - Gère l'espace de noms du système de fichiers.
 - Régule l'accès du client aux fichiers.
 - Il exécute également des opérations sur le système de fichiers telles que renommer, fermer et ouvrir des fichiers et des répertoires.
 - DataNode : C'est un matériel standard doté du système d'exploitation GNU / Linux et du logiciel de datanode. Pour chaque nœud (matériel / système de marchandise) d'un cluster, il y aura un nœud de données. Ces nœuds gèrent le stockage des données de leur système.
 - Les datanodes effectuent des opérations de lecture-écriture sur les systèmes de fichiers, conformément à la demande du client.
 - Ils effectuent également des opérations telles que la création, la suppression et la réplification de blocs conformément aux instructions du namenode.
 - Block : C'est la quantité minimale de données que HDFS peut lire ou écrire est appelée un bloc. La taille de bloc par défaut est de 64 Mo, mais elle peut être augmentée en fonction de la nécessité de modifier la configuration HDFS.

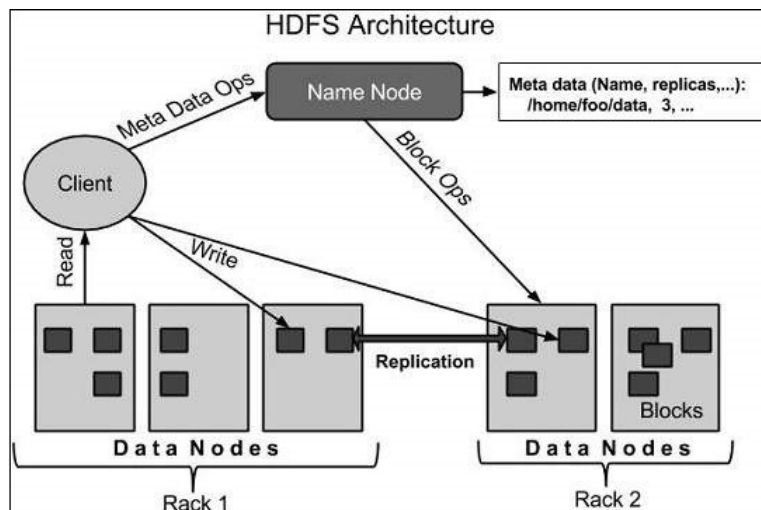


FIGURE 1.6 – HDFS

3. **Hadoop Common** : Il s'agit des bibliothèques Java et des utilitaires requis par d'autres modules Hadoop. Elle fait référence à la collection d'utilitaires et de bibliothèques communs qui prennent en charge d'autres modules Hadoop. Il s'agit d'une partie ou d'un module essentiel du framework Apache Hadoop, avec le système de fichiers distribués Hadoop (HDFS), Hadoop YARN et Hadoop MapReduce.



FIGURE 1.7 – Common

4. **YARN** : Ceci est un cadre pour la planification des travaux et la gestion des ressources de cluster. YARN permet à Hadoop d'étendre et de gérer des milliers de nœuds et de clusters.

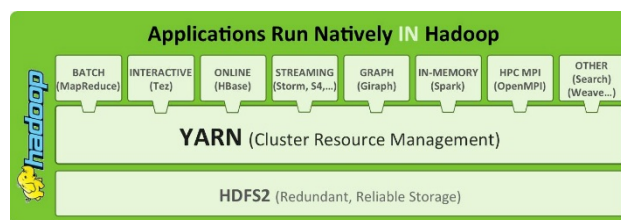


FIGURE 1.8 – Yarn

1.2.3 Avantage de Hadoop

Malgré l'émergence d'options alternatives, en particulier dans le cloud, Hadoop est toujours une technologie importante et précieuse pour les utilisateurs de Big Data pour les raisons suivantes :

- Il peut stocker et traiter rapidement de grandes quantités de données structurées, semi-structurées et non structurées.
- Il protège les applications et le traitement des données contre les pannes matérielles. Si un nœud d'un cluster tombe en panne, les travaux de traitement sont automatiquement redirigés vers d'autres nœuds pour garantir l'exécution continue des applications.
- Il ne nécessite pas que les données soient prétraitées avant d'être stockées. Les organisations peuvent stocker des données brutes dans HDFS et décider ultérieurement comment les traiter et les filtrer pour des utilisations analytiques spécifiques.
- Il est évolutif, de sorte que les entreprises peuvent facilement ajouter plus de nœuds pour permettre à leurs systèmes de gérer plus de données.
- Il peut prendre en charge l'analyse en temps réel pour aider à conduire une meilleure prise de décision opérationnelle, ainsi que des charges de travail par lots pour l'analyse historique.

1.2.4 Inconvénients de Hadoop

Malgré sa popularité, Hadoop est toujours une technologie émergente, et bon nombre de ses limites sont liées à sa nouveauté. Les sous-produits de l'expansion et de l'évolution rapides de Hadoop comprennent des lacunes en matière de compétences, un manque de solutions complémentaires pour répondre à des besoins spécifiques (par exemple, des outils de développement et de débogage, une prise en charge native de Hadoop dans des solutions logicielles spécifiques, etc.). D'autres critiques proviennent du statut de Hadoop en tant que projet open source, car certains professionnels considèrent l'open source trop instable pour les entreprises. D'autres critiques disent que Hadoop est meilleur pour stocker et agréger des données que pour les traiter.

- Exigences de stockage - La redondance intégrée de Hadoop duplique les données, nécessitant ainsi plus de ressources de stockage.
- Prise en charge SQL limitée - Hadoop n'a pas certaines des fonctions de requête auxquelles les utilisateurs de bases de données SQL sont habitués.
- Sécurité native limitée - Hadoop ne crypte pas les données pendant le stockage ou sur le réseau. De plus, Hadoop est basé sur Java, qui est une cible fréquente pour les logiciels malveillants et autres hacks.
- Limitations des composants - Il existe de nombreuses critiques spécifiques concernant les limitations des quatre composants principaux de Hadoop (HDFS, YARN, MapReduce et Common). Certaines de ces limitations sont surmontées par des solutions tierces, mais la fonctionnalité fait défaut dans Hadoop lui-même.

1.2.5 Hadoop et Son ecosysteme

Hadoop est en passe de devenir le standard de Facto de traitement de données, un peu comme Excel est progressivement devenu le logiciel par défaut d'analyse de données. A la différence d'Excel, Hadoop n'a pas été conçu pour être utilisé par les « Analystes métier », mais par les développeurs. Or, l'adoption à grande échelle et le succès d'un standard ne dépendent pas des développeurs, mais des analystes métier. Pour cette raison, les problématiques de la Big Data ont été segmentées d'un point de vue fonctionnel et pour chaque segment, des technologies qui s'appuient sur Hadoop ont été développées pour répondre à ses challenges. L'ensemble de ces outils forment ce qui s'appelle l'écosystème Hadoop. L'écosystème Hadoop enrichit Hadoop et le rend capable de résoudre une grande variété de problématiques métiers. A ce jour, l'écosystème Hadoop est composé d'une centaines de technologies que nous avons choisis de regrouper en 14 catégories selon leur segment de problématique : : les langages d'abstraction, le SQL sur Hadoop (Hive, Pig), les modèles de calcul (MapReduce, Tez), les outils de traitement temps réel (Storm, Spark Streaming), les Bases de données (HBase, Cassandra), les outils d'ingestion streaming (Kafka, Flume), les outils d'intégration des données, (Sqoop, Talend), les outils de coordination de Workflow (Oozie, Control M for Hadoop), les outils de coordination de services distribués (Zookeeper), les outils d'administration de cluster (Ranger, Sentry), les outils d'interface utilisateur (Hue, Jupyter), les outils d'indexation de contenu (ElasticSearch, Splunk), les systèmes de fichier distribués (HDFS), et les gestionnaires de ressources (YARN et MESOS). Dans cette partie, nous allons passer en revue la fonction de chacun des outils qui constituent cet écosystème de technologies Big Data. Après, si vous souhaitez aller plus loin, nous vous recommandons de télécharger notre guide "Initiation à l'écosystème Hadoop" qui juste situé à votre droite. La carte heuristique suivante présente de façon globale l'écosystème Hadoop.

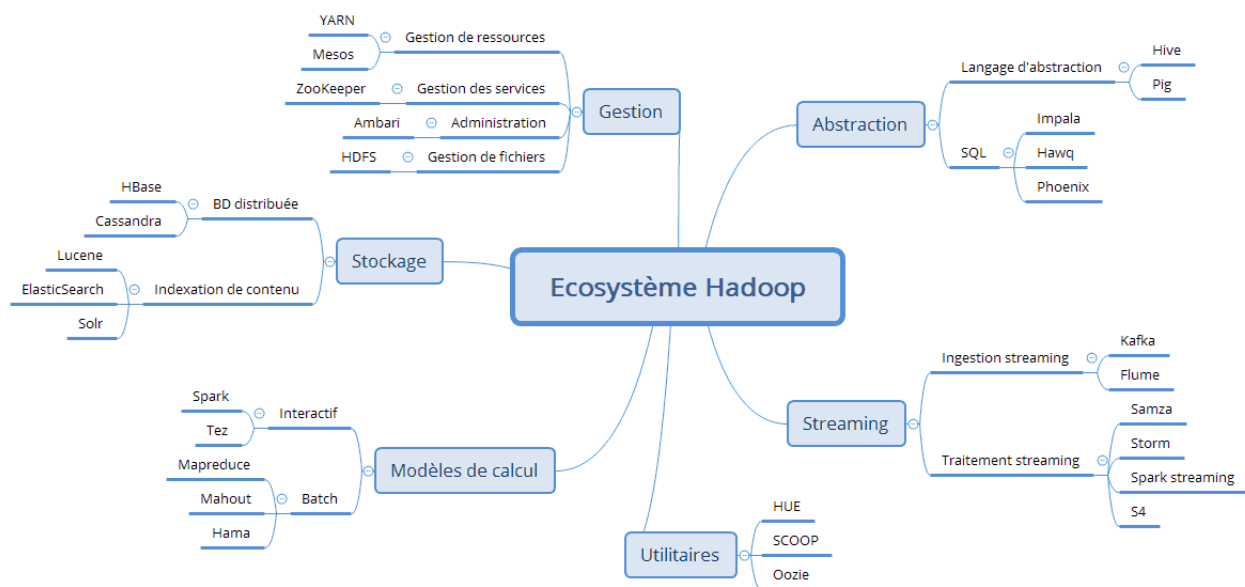


FIGURE 1.9 – Ecosysteme Hadoop

La configuration de base de l'écosystème Hadoop contient les technologies suivantes : Spark, Hive, PIG, HBase, Sqoop, Storm, ZooKeeper, Oozie et Kafka.

— **SPARK**

Avant d'expliquer ce que c'est que Spark, rappelons que pour qu'un algorithme puisse s'exécuter sur plusieurs nœuds d'un cluster Hadoop, il faut qu'il soit parallélisable. Ainsi, on dit d'un algorithme qu'il est "scalable" s'il est parallélisable (et peut donc profiter de la scalabilité d'un cluster). Hadoop est une implémentation du modèle de calcul MapReduce. Le problème avec le MapReduce est qu'il est bâti sur un modèle de Graphe Acyclique Direct. En d'autres termes, l'enchaînement des opérations du MapReduce s'exécutent en trois phases séquentielles directes et sans détour (Map -> Shuffle -> Reduce), aucune phase n'est itérative (ou cyclique). Le modèle acyclique direct n'est pas adapté à certaines applications, notamment celles qui réutilisent les données à travers de multiples opérations, telles que la plupart des algorithmes d'apprentissage statistique, itératifs pour la plupart, et les requêtes interactives d'analyse de données. Spark est une réponse à ces limites, c'est un moteur de calcul qui effectue des traitements distribués en mémoire sur un cluster. Autrement dit, c'est un moteur de calcul in-memory distribué. Comparativement au MapReduce qui fonctionne en mode batch, le modèle de calcul de Spark fonctionne en mode interactif, c'est à dire, monte les données en mémoire avant de les traiter et est de ce fait très adapté au traitement de Machine Learning.



FIGURE 1.10 – Spark

— **HIVE**

Hive est une infrastructure informatique similaire au Data Warehouse qui fournit des services de requêtes et d'agrégation de très gros volumes de données stockées sur un système de fichier distribué de type HDFS. Hive fournit un langage de requête basé sur le SQL (norme ANSI-92) appelé HiveQL (Hive Query Language), qui est utilisé pour adresser des requêtes aux données stockées sur le HDFS. Le HiveQL permet également aux utilisateurs avancés/développeurs d'intégrer des fonctions Map et Reduce directement à leurs requêtes pour couvrir une plus large palette de problèmes de gestion de données. Lorsque vous écrivez une requête en HiveQL, cette requête est transformée en job MapReduce et soumise au JobTracker pour exécution par Hive. Pour plus de détails sur Hive, consultez le tutoriel suivant : <https://www.data-transitionnumerique.com/le-sql-dans-hadoop-hive-pig/>

Apache Hive



FIGURE 1.11 – Hive

— **PIG**

Pig est un environnement d'exécution de flux interactifs de données sous Hadoop. Il est composé de 2 éléments : un langage d'expression de flux de données appelé le Pig Latin ; et un environnement Interactif d'exécution de ces flux de données ;

Le langage offert par Pig, le Pig Latin, est à peu près similaire au langage de Scripting tels que Perl, Python, ou Ruby. Cependant, il est plus spécifique que ces derniers et se décrit mieux sur le terme "langage de flux de données" (data flow language). Il permet d'écrire des requêtes sous forme de flux séquentiels de données source pour obtenir des données « cible » sous Hadoop à la façon d'un ETL. Ces flux sont ensuite transformés en fonctions MapReduce qui sont enfin soumises au jobtracker pour exécution. Pour faire simple, Pig c'est l'ETL d'Hadoop. Programmer en Pig Latin revient à décrire sous forme de flux indépendants mais imbriqués, la façon dont les données sont chargées, transformées, et agrégées à l'aide d'instructions Pig spécifiques appelées opérateurs. La maîtrise de ces opérateurs est la clé de la maîtrise de la programmation en Pig Latin, d'autant plus qu'ils ne sont pas nombreux relativement au Hive par exemple. Pour plus de détails sur Pig, consultez le tutoriel suivant : <https://www.data-transitionnumerique.com/le-sql-dans-hadoop-hive-pig/>



FIGURE 1.12 – Pig

— **HBASE**

Avant de parler de HBase, nous allons rappeler que les SGBDR, qui sont jusqu'à présent utilisés pour la gestion des données ont montré très rapidement leurs limites face d'une part la forte volumétrie des données et d'autre part face à la diversité des données. En effet, les SGBDR sont conçus pour gérer uniquement des données structurées (table de données en ligne/colonnes), de plus l'augmentation du volume des données augmente le temps de latence des requêtes. Cette latence est préjudiciable dans le cadre de nombreux métiers requérant des réponses en temps quasi-réel. Pour répondre à ces limites, de nouveaux SGBD dit "NoSQL" ont vu le jour. Ceux-ci n'imposent pas de structure particulière aux données, sont capables de distribuer le stockage et la gestion des données sur plusieurs nœuds et sont scalables. HBase est un SGBD distribué, orienté-colonne qui fournit l'accès en temps réel aussi bien en lecture qu'en écriture aux données stockées sur le HDFS. Là où le HDFS fournit un accès séquentiel aux données en batch, non-approprié pour des problématiques d'accès rapide à la donnée comme le Streaming, HBase couvre ces lacunes et offre un accès rapide aux données stockées sur le HDFS.

Il a été conçu à partir du SGBD de Google "Big Table" et est capable de stocker de très grosses volumétries de données (milliard de lignes/colonnes). Il dépend de ZooKeeper, un service de coordination distribuée pour le développement d'applications.



FIGURE 1.13 – HBase

– SGOOP

Sqoop ou SQL-to-Hadoop est un outil qui permet de transférer les données d'une base de données relationnelle au HDFS d'Hadoop et vice-versa. Il est intégré à l'écosystème Hadoop et est ce que nous appelons le planificateur d'ingestion des données dans Hadoop. Vous pouvez utiliser Sqoop pour importer des données des SGBDR tels que MySQL, Oracle, ou SQL Server au HDFS, transformer les données dans Hadoop via le MapReduce ou un autre modèle de calcul, et les exporter en retour dans le SGBDR. Nous l'appelons planificateur d'ingestion des données parce que tout comme Oozie (plus bas), il automatise ce processus d'import/export et en planifie le moment d'exécution. Tout ce que vous avez à faire en tant qu'utilisateur c'est d'écrire les requêtes SQL qui vont être utilisées pour effectuer le mouvement d'import/export. Par ailleurs, Sqoop, utilise le MapReduce pour importer et exporter les données, ce qui efficace et tolérant aux pannes. La figure suivante illustre particulièrement bien les fonctions de Sqoop. Vous pouvez lire le tutoriel suivant pour voir comment Sqoop se positionne devant des ETL comme Talend : [https ://www.data-transitionnumerique.com/sqoop-vs-talend/](https://www.data-transitionnumerique.com/sqoop-vs-talend/)



FIGURE 1.14 – Sqoop

– STORM

Pour comprendre Storm, il faut comprendre la notion d'architectures lambda et pour comprendre l'intérêt des architectures lambda, il faut comprendre le concept d'objets connectés. Les objets connectés ou Internet des objets (IoT – Internet of Things en anglais) représente l'extension d'Internet à nos vies quotidiennes. Elle génère des données en streaming et dans la plupart de ses problématiques, nécessite que les données soient traitées en temps réel. Les modèles que vous connaissez tels que les modèles de calcul Batch ne sont pas adaptés aux problématiques temps réel que soulève l'IoT. Même les modèles de calcul interactif ne sont pas adaptés pour faire du traitement continu en temps réel. A la différence des données opérationnelles produites par les systèmes opérationnels d'une entreprise comme la finance, le marketing, qui même lorsqu'elles sont produites en streaming peuvent être historisées pour un traitement ultérieur, les données produites en streaming dans le cadre des phénomènes comme l'IoT ou Internet se périment (ou ne sont plus valides) dans les instants qui suivent leur création et exigent donc un traitement immédiat. En dehors des objets connectés, les problématiques métier comme la lutte contre la fraude, l'analyse des données de réseau sociaux, la géolocalisation, exigent des temps de réponse très faibles, quasiment de l'ordre de moins d'une seconde. Pour résoudre cette problématique dans un contexte Big Data, des architectures dites lambda ont été mises sur pieds. Ces architectures ajoutent au MapReduce 2 couches de traitements supplémentaires pour la réduction des temps de latence. Storm est une implémentation logicielle de l'architecture lambda. Il permet de développer sous Hadoop des applications qui traitent les données en temps réel (ou presque).



FIGURE 1.15 – Storm

- **ZOOKEEPER** La synchronisation ou coordination de la communication entre les nœuds lors de l'exécution des tâches parallèles est l'un des problèmes les plus difficiles dans le développement d'application distribuée. Pour résoudre ce problème, Hadoop a introduit dans son écosystème des outils de coordination de service, en l'occurrence ZooKeeper. ZooKeeper prend en charge la complexité inhérente de la synchronisation de l'exécution des tâches distribuées dans le cluster et permet aux autres outils de l'écosystème Hadoop de ne pas avoir à gérer ce problème eux-mêmes. Il permet également aux utilisateurs de pouvoir développer des applications distribuées sans être des experts de la programmation distribuée. Sans entrer dans les détails complexes de la coordination des données entre les nœuds d'un cluster Hadoop, ZooKeeper fournit un service de configuration distribué, un service de distribution et un registre de nommage pour les applications distribuées. ZooKeeper est le moyen utilisé par Hadoop pour coordonner les jobs distribués.



FIGURE 1.16 – Zookeeper

- **OOZIE** Par défaut, Hadoop exécute les jobs au fur et à mesure qu'ils sont soumis par l'utilisateur sans tenir compte de la relation qu'ils peuvent avoir les uns avec les autres. Or, les problématiques pour lesquelles l'on utilise Hadoop demandent généralement la rédaction d'un ou de plusieurs jobs complexes. Lorsque les 2 jobs seront soumis au JobTracker (ou à YARN) par exemple, celui-ci va les exécuter sans faire attention au lien qui existe entre eux, ce qui risque de causer une erreur (exception) et entraîner l'arrêt du code. Comment fait-on pour gérer l'exécution de plusieurs jobs qui sont relatifs au même problème ? Pour gérer ce type de problème, la solution la plus simple actuellement consiste à utiliser un planificateur de jobs, en l'occurrence Oozie. Oozie est un planificateur d'exécution des jobs qui fonctionne comme un service sur un cluster Hadoop. Il est utilisé pour la planification des jobs Hadoop, et plus généralement pour la planification de l'exécution de l'ensemble des jobs qui peuvent s'exécuter sur un cluster, par exemple un script Hive, un job MapReduce, un job Hama, un job Storm, etc. Il a été conçu pour gérer l'exécution immédiate, ou différée de milliers de jobs interdépendants sur un cluster Hadoop automatiquement. Pour utiliser Oozie, il suffit de configurer 2 fichiers XML : un fichier de configuration du moteur Oozie et un fichier de configuration du workflow des jobs.



FIGURE 1.17 – Oozie

1.2.6 Pourquoi la sécurité Hadoop est-elle importante ?

- Lois régissant la confidentialité des données : particulièrement importantes pour les secteurs des soins de santé et des finances
- Réglementation du contrôle des exportations d'informations de défense
- Protection des données de recherche exclusives
- Politiques de l'entreprise
- Différentes équipes dans une entreprise ont des besoins différents
- La configuration de plusieurs clusters est une solution courante : un cluster peut contenir des données protégées, un autre ne

1.2.7 Les trois A de la sécurité et de la protection des données

Que faire de la sécurité Hadoop ? En partie, la sécurité et la gouvernance dans Hadoop nécessitent bon nombre des mêmes approches que dans le monde traditionnel de la gestion des données. Il s'agit notamment des «trois en tant que» de la sécurité et de la protection des données.

1. **Autorisation** : c'est le processus de détermination des données, types de données ou applications auxquels l'utilisateur est autorisé à accéder.
 - Déterminer si un participant est autorisé à effectuer une action
 - Généralement effectué en vérifiant une liste de contrôle d'accès
2. **Authentification** : il s'agit simplement de déterminer avec précision l'identité d'un utilisateur donné tentant d'accéder à un Cluster ou application Hadoop basé sur l'un des nombreux facteurs.
 - Confirmer l'identité d'un participant
 - Généralement effectué en vérifiant les informations d'identification (nom d'utilisateur / mot de passe)
3. **Audit** : c'est le processus d'enregistrement et de rapport de ce qu'un utilisateur autorisé authentifié a accordé une fois l'accès au cluster, y compris les données qui ont été consultées / modifiées / ajoutées et quelle analyse s'est produite.

Protection des données : la protection des données se réfère à l'utilisation de techniques telles que le cryptage et le masquage des données pour empêcher l'accès aux données sensibles par des utilisateurs et des applications non autorisés.

1.2.8 Types de sécurité Hadoop

- Conception de sécurité Hadoop : kerberos
- Mise en place d'un cluster Hadoop sécurisé : Configurer Hadoop avec l'authentification kerberos
- Sécuriser l'écosystème Hadoop : Configurer Kerberos pour les composants de l'écosystème Hadoop
- Intégration de Hadoop avec les systèmes de sécurité d'entreprise :
- Sécurisation des données sensibles à Hadoop :
- Sécurité Événement et audit Logging à Hadoop : Incident de sécurité et surveillance des événements dans un cluster Hadoop

Chapitre 2

Installation de Hadoop

2.1 Mode de Fonctionnement

Une fois que vous avez téléchargé Hadoop, vous pouvez utiliser votre cluster Hadoop dans l'un des trois modes pris en charge :

- **Mode local / autonome** - Après avoir téléchargé Hadoop dans votre système, par défaut, il est configuré en mode autonome et peut être exécuté comme un processus java unique.
- **Mode pseudo distribué** - Il s'agit d'une simulation distribuée sur une seule machine. Chaque démon Hadoop tel que hdfs, yarn, MapReduce etc., s'exécutera comme un processus java distinct. Ce mode est utile pour le développement.
- **Mode entièrement distribué** - Ce mode est entièrement distribué avec au moins deux machines ou plus en tant que cluster. Nous verrons ce mode en détail dans les prochains chapitres.

2.2 Oracle Big Data Lite

Ils existent plusieurs distributions de linux , Pour pouvoir faire nos tests de configuration nous avons utilisé une machine virtuelle nommée **oracle big data lite** pour installer Hadoop

Oracle Big Data Lite Virtual Machine fournit un environnement intégré pour vous aider à démarrer avec la plate-forme Oracle Big Data. De nombreux composants de la plateforme Oracle Big Data ont été installés et configurés, ce qui vous permet de commencer à utiliser le système immédiatement.

Les composants suivants sont inclus dans Oracle Big Data Lite :

- Oracle Enterprise Linux 6.9
- Oracle Database 12c Release 1 Enterprise Edition (12.1.0.2) - y compris les tables externes compatibles Oracle Big Data SQL, Oracle Multitenant, Oracle Advanced Analytics, Oracle OLAP, Oracle Partitioning, Oracle Spatial and Graph, etc.
- Distribution de Cloudera avec Apache Hadoop (CDH5.13.1)
- Gestionnaire Cloudera (5.13.1)
- Oracle Big Data Spatial and Graph 2.4
- Connecteurs Oracle Big Data 4.11
- Oracle SQL Connector pour HDFS 3.8.1
- Oracle Loader pour Hadoop 3.9.1
- Oracle Data Integrator 12c (12.2.1.3.0)
- Oracle R Advanced Analytics pour Hadoop 2.7.1
- Oracle XQuery pour Hadoop 4.9.1

- Source de données Oracle pour Apache Hadoop 1.2.1
- Oracle Shell pour les chargeurs Hadoop 1.3.1
- Oracle NoSQL Database Enterprise Edition 12cR1 (4.5.12)
- Oracle JDeveloper 12c (12.2.1.2.0)
- Oracle SQL Developer et Data Modeler 17.3.1 avec Oracle REST Data Services 3.0.7
- Oracle Data Integrator 12cR1 (12.2.1.3.0)
- Oracle GoldenGate 12c (12.3.0.1.2)
- Oracle R Distribution 3.3.0
- Oracle Perfect Balance 2.10.0

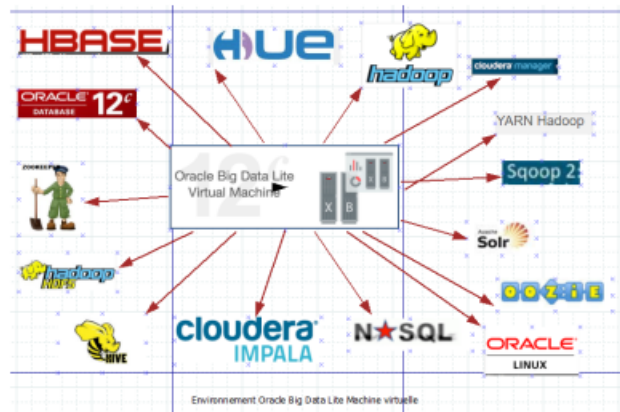


FIGURE 2.1 – Big Data lite

2.2.1 Installation

1. Telecharger VirtualBox



FIGURE 2.2 – Virtualbox

2. Telecharger Oracle Big Data lite





 BigDataLite411.7z.001 (2147483648 octets)
 BigDataLite411.7z.002 (2147483648 octets)
 BigDataLite411.7z.003 (2147483648 octets)
 BigDataLite411.7z.004 (2147483648 octets)
 BigDataLite411.7z.005 (2147483648 octets)
 BigDataLite411.7z.006 (2147483648 octets)
 BigDataLite411.7z.007 (2147483648 octets)
 BigDataLite411.7z.008 (2147483648 octets)
 BigDataLite411.7z.009 (2147483648 octets)
 BigDataLite411.7z.010 (2147483648 octets)
 BigDataLite411.7z.011 (2147483648 octets)
 BigDataLite411.7z.012 (27131890 octets)

FIGURE 2.3 – Telechargement Big Data lite

3. Executer Big data lite







 7z1900-x64	22/02/2019 09:30	Application	1 414 Ko
 BigDataLite411	12/01/2018 17:22	Open Virtualizatio...	23 446 964...
 bigdatalite-quickdeploy-411-4219565	28/11/2020 12:18	Foxit Reader PDF ...	763 Ko
 Oracle_VM_VirtualBox_Extension_Pack-6....	04/09/2020 12:07	VirtualBox Extensi...	10 885 Ko
 p7zip_16.02_src_all.tar.bz2	14/07/2016 09:39	Fichier BZ2	4 141 Ko
 virtualbox-6.1_6.1.14-140239~Ubuntu~bi...	04/09/2020 09:45	Fichier DEB	86 063 Ko

FIGURE 2.4 – Execution Hadoop

4. Demarrer la machine Machine virtuelle Oracle Big Data Lite

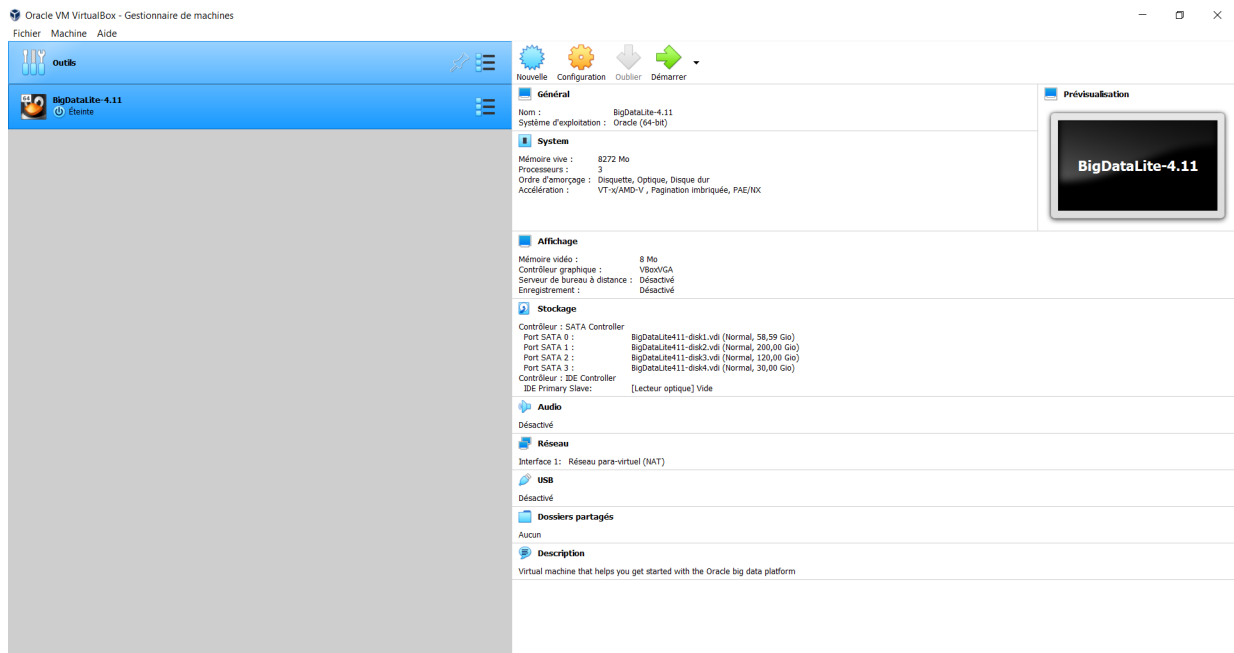


FIGURE 2.5 – Demarrage de la machine

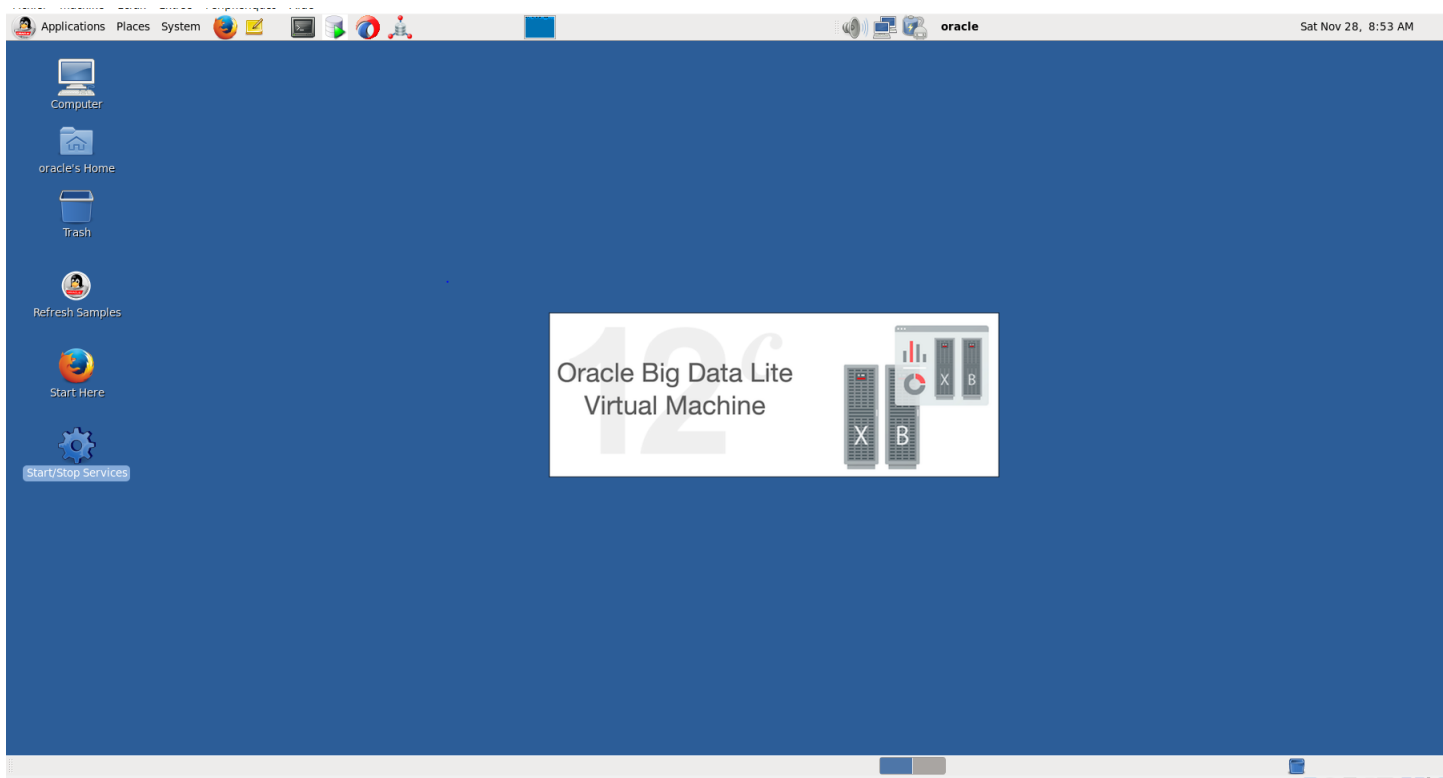


FIGURE 2.6 – Big Data lite 2

Se connecter avec :

id : oracle

password : welcome1

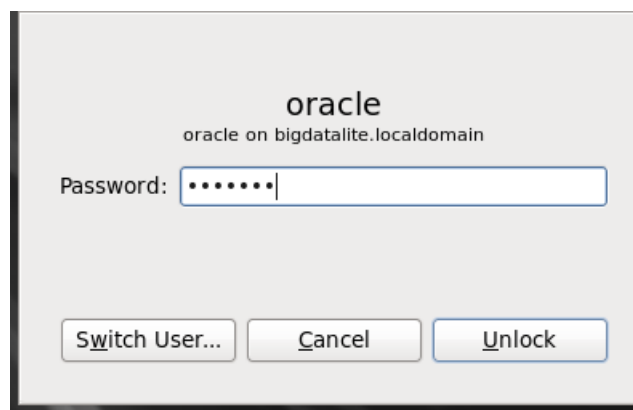


FIGURE 2.7 – connection Big Data lite

2.2.2 Configuration

Pour commencer nous devons démarrer les services dont on aura besoin pour faire nos tests à savoir : **HDFS ,YARN**

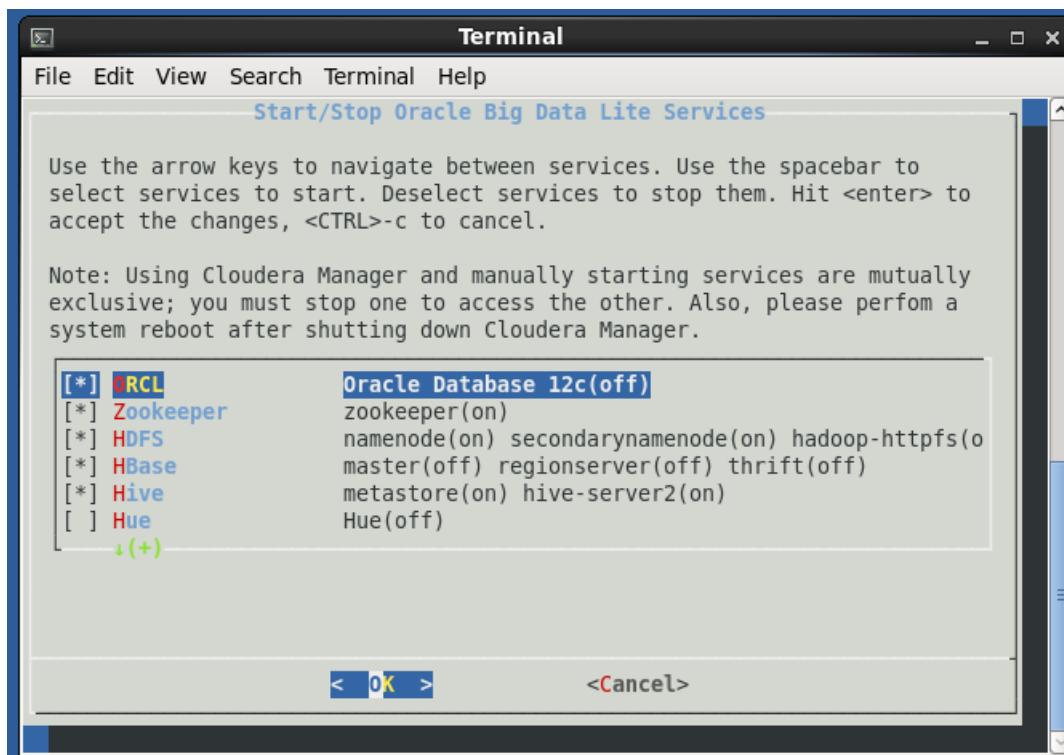


FIGURE 2.8 – Service Big Data lite

Pour démarrer le serveur Cloudera Manager :**sudo service cloudera-scm-server start**

```
[oracle@bigdatalite ~]$ sudo service cloudera-scm-server start
Starting cloudera-scm-server: [ OK ]
[oracle@bigdatalite ~]$ sudo service cloudera-scm-server status
cloudera-scm-server (pid 5512) is running...
```

FIGURE 2.9 – Demarrage de Cloudera

Pour démarrer l'interface CM, nous devons soit taper l'adresse manuellement, soit sélectionner «Cloudera Manager» dans la barre d'outils des signets du navigateur. Après le lancement de l'interface CM, le résumé des services est présenté dans la figure 1 ci-dessous.(localhost :7180/) **Se**

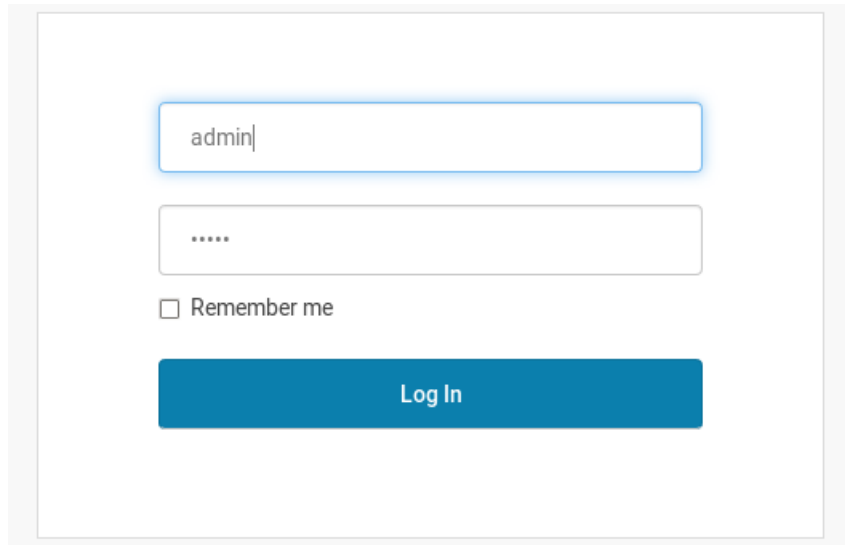
A login form for Cloudera Manager. It features a text input field containing the username 'admin', a password input field with masked characters '*****', a checkbox labeled 'Remember me', and a blue 'Log In' button.

FIGURE 2.10 – Connection Cloudera

connecter avec :

id : admin

password : admin

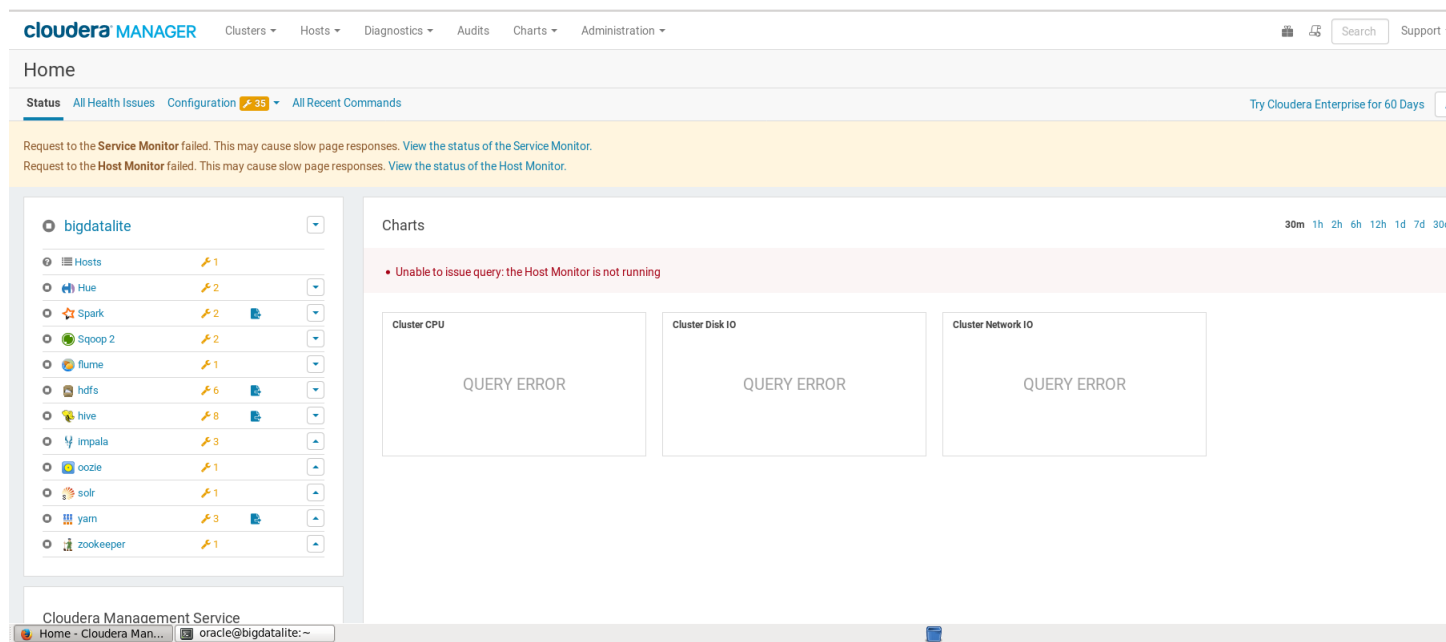


FIGURE 2.11 – Cloudera

2.3 Hortonworks

Ils existent plusieurs distributions de linux , nous allons utiliser Hortonworks pour installer Hadoop car elle fournit tout les nécessité pour Apache 2. **Hortonworks** est une société de logiciels informa-



FIGURE 2.12 – Distribution de linux

tique basée à Santa Clara, en Californie. La société se concentre sur le développement et le soutien de Hadoop, un framework Java qui permet le traitement distribué de grands volumes de données par des grappes de serveurs. Elle a fusionné avec Cloudera.



FIGURE 2.13 – Hortonworks

2.3.1 Installation

Pour l'installation nous allons utiliser Docker et HDP dans une machine linux (**Ubuntu**) pour avoir Hadoop .

- **Docker** est un logiciel libre permettant de lancer des applications dans des conteneurs logiciels.
- **HDP** est un framework open source pour le stockage distribué et le traitement de grands ensembles de données multi-sources. HDP modernise votre infrastructure informatique et sécurise vos données, dans le cloud ou sur site, tout en vous aidant à générer de nouvelles sources de revenus, à améliorer l'expérience client et à contrôler les coûts.



FIGURE 2.14 – Docker + HDP

1. Dans le dossier décompressé, vous trouverez le script shell `docker-deploy.sh` . **sh docker-deploy-30.sh**

```

dj1te@ubuntu:~/Documents/HDP_3.0.1_docker-deploy-scripts_18120587fc7fb$ sudo sh
docker-deploy-(HDPversion).sh
[sudo] password for dj1te:
dj1te@ubuntu:~/Documents/HDP_3.0.1_docker-deploy-scripts_18120587fc7fb$ ls
scripts docker-deploy-hdp30.sh
dj1te@ubuntu:~/Documents/HDP_3.0.1_docker-deploy-scripts_18120587fc7fb$ sudo sh
docker-deploy-hdp30.sh
[sudo] password for dj1te:
+ registry=hortonworks
+ name=sandbox-hdp
+ version=3.0.1
+ proxyName=sandbox-proxy
+ proxyVersion=1.0
+ flavor=hdp
+ echo hdp
+ mkdir -p sandbox/proxy/conf.d
+ mkdir -p sandbox/proxy/conf.stream.d
+ docker pull hortonworks/sandbox-hdp:3.0.1
3.0.1: Pulling from hortonworks/sandbox-hdp
70799bbf2220: Pulling fs layer
40963917cdad: Pulling fs layer
3fe9adb8d87e: Pulling fs layer
e03ec4e8c3d0: Pulling fs layer
7ea5917732c0: Pulling fs layer

```

2. Vérifiez que le sandbox HDP a été déployé avec succès : **docker ps**

```

dj1te@ubuntu:~/Documents/HDP_3.0.1_docker-deploy-scripts_18120587fc7fb$ sudo docker start sandbox-hdp
[sudo] password for dj1te:
sandbox-hdp
dj1te@ubuntu:~/Documents/HDP_3.0.1_docker-deploy-scripts_18120587fc7fb$ sudo docker start sandbox-proxy
sandbox-proxy
dj1te@ubuntu:~/Documents/HDP_3.0.1_docker-deploy-scripts_18120587fc7fb$ sudo docker ps
CONTAINER ID        IMAGE                                     COMMAND                  CREATED             STATUS
US
PORTS
NAMES
6ed2534ff3b8      hortonworks/sandbox-proxy:1.0          "nginx -g 'daemon of..." 0 hours ago         Up 4
seconds
0.0.0.0:1000->1000/tcp, 0.0.0.0:1100->1100/tcp, 0.0.0.0:1111->1111/tcp, 0.0.0.0:1900-
>1900/tcp, 0.0.0.0:2100->2100/tcp, 0.0.0.0:2181-2182->2181-2182/tcp, 0.0.0.0:2201-2202->2201-2202/tcp
, 0.0.0.0:2222->2222/tcp, 0.0.0.0:3000->3000/tcp, 0.0.0.0:4040->4040/tcp, 0.0.0.0:4200->4200/tcp, 0.0
.0.0:4242->4242/tcp, 0.0.0.0:4557->4557/tcp, 0.0.0.0:5007->5007/tcp, 0.0.0.0:5011->5011/tcp, 0.0.0.0:
6001->6001/tcp, 0.0.0.0:6003->6003/tcp, 0.0.0.0:6006->6006/tcp, 0.0.0.0:6080->6080/tcp, 0.0.0.0:6188-
>6188/tcp, 0.0.0.0:6627->6627/tcp, 0.0.0.0:6657-6668->6657-6668/tcp, 0.0.0.0:7777->7777/tcp, 0.0.0.0:
7788->7788/tcp, 0.0.0.0:8000->8000/tcp, 0.0.0.0:8005->8005/tcp, 0.0.0.0:8020->8020/tcp, 0.0.0.0:8032-
>8032/tcp, 0.0.0.0:8040->8040/tcp, 0.0.0.0:8042->8042/tcp, 0.0.0.0:8080-8082->8080-8082/tcp, 0.0.0.0:
8086->8086/tcp, 0.0.0.0:8088->8088/tcp, 0.0.0.0:8090-8091->8090-8091/tcp, 0.0.0.0:8188->8188/tcp, 0.0

```

2.3.2 Configuration

3. Pour voir l'adresse ip du centenaire sandbox-hdp **docker inspect -f 'range .NetworkSettings.Networks.IPAddressend' sandbox-hdp**

```

root@ubuntu:~# docker inspect -f '{{range .NetworkSettings.Networks}}{{.IPAddress}}{{end}}' sandbox-hdp
172.17.0.2

```

4. Mappez l'IP Sandbox à votre nom d'hôte souhaité dans le fichier Hosts **echo '172.17.0.2 sandbox-hdp.hortonworks.com sandbox-hdf.hortonworks.com' | sudo tee -a /etc/hosts**

```

root@ubuntu:~# echo '172.17.0.2 sandbox-hdp.hortonworks.com sandbox-hdf.hortonworks.com' | sudo tee -a /etc/hosts
172.17.0.2 sandbox-hdp.hortonworks.com sandbox-hdf.hortonworks.com

```


2.3.3 Access Terminal

Identifiant (s) de nom	Rôle	Prestations de service
Sam Admin	Administrateur Ambari	Ambari
Raj (raj_ops)	Opérateur d'entrepôt Hadoop	Ruche / Tez, Ranger, Falcon, Knox, Sqoop, Oozie, Flume, Zookeeper
Maria (maria_dev)	Développeur Spark et SQL	Hive, Zeppelin, MapReduce / Tez / Spark, Pig, Solr, HBase / Phoenix, Sqoop, NiFi, Storm, Kafka, Flume
Amy (amy_ds)	Data Scientist	Spark, Hive, R, Python, Scala
Holger (holger_gov)	Responsable des données	Atlas

Autorisation au niveau du système d'exploitation

Identifiant (s) de nom	Autorisation HDFS	Autorisation Ambari	Autorisation Ranger
Sam Admin	Opérations maximales	Administrateur Ambari	Accès administrateur
Raj (raj_ops)	Accès à Hive, Hbase, Atlas, Falcon, Knox, Sqoop, Oozie, Flume, Operations	Administrateur de cluster	Accès administrateur
Maria (maria_dev)	Accès à Hive, Hbase, Falcon, Oozie et Spark	Opérateur de service	Accès utilisateur normal
Amy (amy_ds)	Accès à Hive, Spark et Zeppelin	Opérateur de service	Accès utilisateur normal
Holger (holger_gov)	Accès à Atlas	Administrateur de service	Accès utilisateur normal

http://sandbox-hdp.hortonworks.com :4200/ ou ssh root@sandbox-hdp.hortonworks.com -p 2222

```

Ambari
root@sandbox-hdp:~ - ssh X
sandbox-hdp.hortonworks.com:4200
sandbox-hdp login: root
root@sandbox-hdp.hortonworks.com's password:
You are required to change your password immediately (root enforced)
Last login: Sat Oct 17 15:54:05 2020
Changing password for root.
(current) UNIX password:
New password:
BAD PASSWORD: The password fails the dictionary check - it is too simplistic/systematic
New password:
Retype new password:
[root@sandbox-hdp ~]#

```

```

[root@sandbox-hdp ~]# sandbox-version
== Sandbox Information ==
Platform: hdp-security
Build date: 11-29-2018
Ambari version: 2.7.1.0-169
Hadoop version: Hadoop 3.1.1.3.0.1.0-187
OS: CentOS Linux release 7.5.1804 (Core)
=====

```

2.3.4 Page d'accueil

La page d'accueil de Sandbox est également connue sous le nom de page d'accueil . Il fonctionne sur le numéro de port : 1080 . Pour l'ouvrir, utilisez votre adresse d'hôte et ajoutez le numéro de port **http://sandbox-hdp.hortonworks.com:1080/**

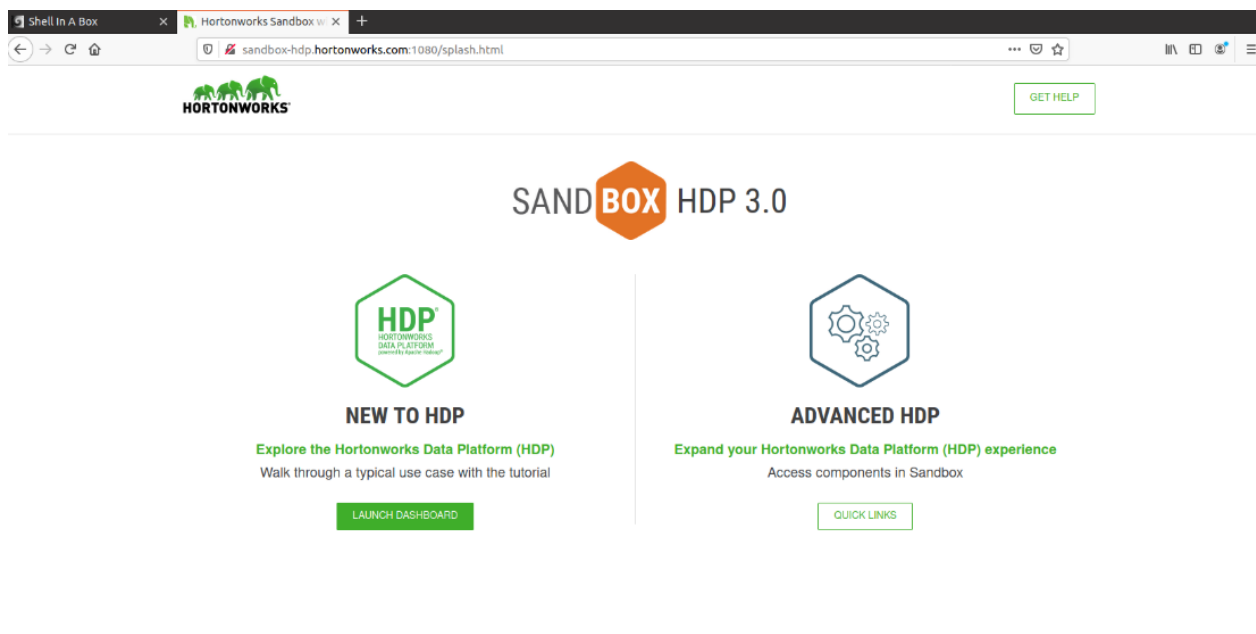


FIGURE 2.15 – Accueil hortonworks

2.3.5 Ambari : Définition et Réinitialisation du mot de passe administrateur

1. Définition

Le projet Apache Ambari vise à simplifier la gestion Hadoop en développant des logiciels pour le provisionnement, la gestion et la surveillance des clusters Apache Hadoop. Ambari fournit une interface utilisateur Web de gestion Hadoop intuitive et facile à utiliser, soutenue par ses API RESTful. Ambari permet aux administrateurs système de :

- Provisionner un cluster Hadoop
 - Ambari fournit un assistant pas à pas pour installer les services Hadoop sur n'importe quel nombre d'hôtes.
 - Ambari gère la configuration des services Hadoop pour le cluster.
- Gérer un cluster Hadoop
 - Ambari fournit une gestion centrale pour démarrer, arrêter et reconfigurer les services Hadoop sur l'ensemble du cluster.
- Surveiller un cluster Hadoop
 - Ambari fournit un tableau de bord pour surveiller la santé et l'état du cluster Hadoop.
 - Ambari utilise le système de métriques Ambari pour la collecte de métriques.
 - Ambari exploite Ambari Alert Framework pour l'alerte système et vous avertit lorsque votre attention est nécessaire (par exemple, un nœud tombe en panne, l'espace disque restant est faible, etc.).
- Ambari permet aux développeurs d'applications et aux intégrateurs système de :
 - Intégrez facilement les capacités de provisionnement, de gestion et de surveillance Hadoop à leurs propres applications avec les API REST Ambari .

Ambari Dashboard fonctionne sur le port : 8080 **http://sandbox-hdp.hortonworks.com:8080**

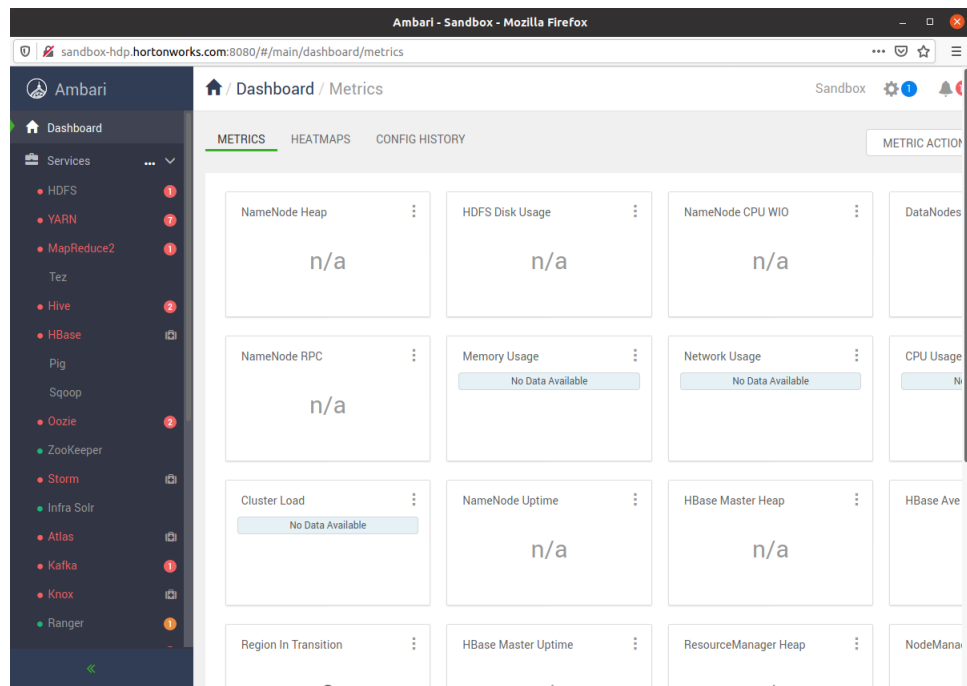
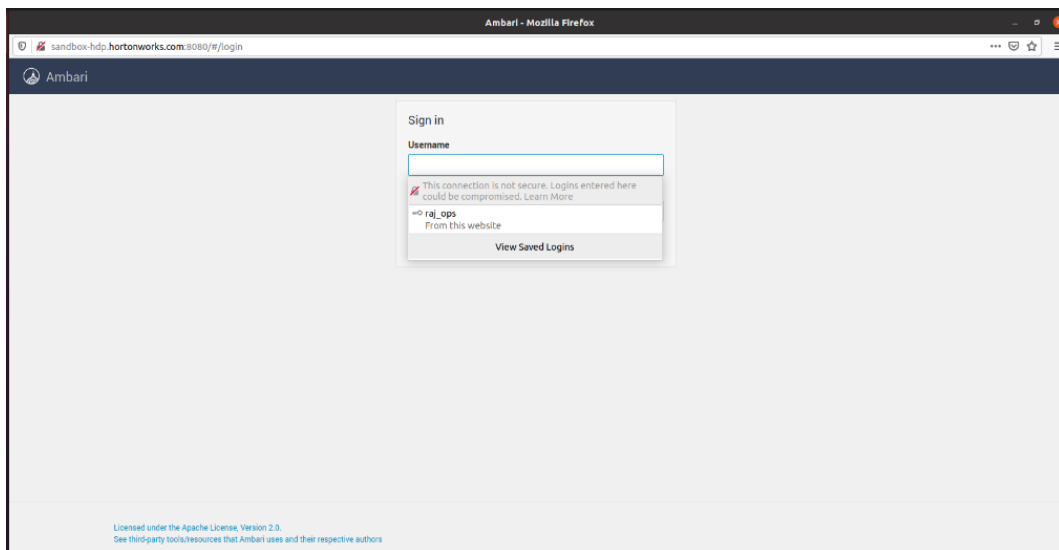


FIGURE 2.16 – Ambari

2. Réinitialisation du mot de passe administrateur

En raison de la possibilité que les mots de passe soient vulnérables au piratage, nous vous recommandons de modifier votre mot de passe administrateur Ambari pour qu'il soit unique.

- (a) Ouvrez le client Web Shell (alias Shell-in-a-Box) :
- (b) La connexion en utilisant les identifiants : root / hadoop
- (c) Tapez les commandes suivantes : ambari-admin-password-reset

```
[root@sandbox-hdp ~]# ambari-admin-password-reset
Please set the password for admin:
Please retype the password for admin:
[root@sandbox-hdp ~]# ambari-admin-password-reset
Please set the password for admin:
Please retype the password for admin:

The admin password has been set.
Restarting ambari-server to make the password change effective...

Using python /usr/bin/python
Restarting ambari-server
Waiting for server stop...
Ambari Server stopped
Ambari Server running with administrator privileges.
Organizing resource files at /var/lib/ambari-server/resources...
Ambari database consistency check started...
Server PID at: /var/run/ambari-server/ambari-server.pid
Server out at: /var/log/ambari-server/ambari-server.out
Server log at: /var/log/ambari-server/ambari-server.log
Waiting for server start.....
Server started listening on 8080

DB configs consistency check: no errors and warnings were found.
[root@sandbox-hdp ~]#
```

FIGURE 2.17 – Reset password ambari

Deuxième partie

Sécurisation de Hadoop

2.4 Autorisation de fichier

Le système de fichiers distribués Hadoop (HDFS) implémente un modèle d'autorisations pour les fichiers et les répertoires qui partage une grande partie du modèle POSIX (Portable Operating System Interface). Chaque fichier et répertoire est associé à un propriétaire et à un groupe. Le fichier ou le répertoire dispose d'autorisations distinctes pour l'utilisateur propriétaire, pour les autres utilisateurs membres du groupe et pour tous les autres utilisateurs. Pour les fichiers, l'autorisation *r* est requise pour lire le fichier et l'autorisation *w* est requise pour écrire ou ajouter au fichier. Pour les répertoires, l'autorisation *r* est requise pour répertorier le contenu du répertoire, l'autorisation *w* est requise pour créer ou supprimer des fichiers ou des répertoires et l'autorisation *x* est requise pour accéder à un enfant du répertoire.

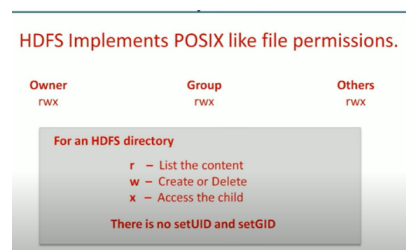


FIGURE 2.18 – POSIX

Chaque processus client qui accède à HDFS possède une identité en deux parties composée du nom d'utilisateur et de la liste des groupes. Chaque fois que HDFS doit effectuer une vérification des autorisations pour un fichier ou un répertoire *foo* auquel un processus client accède,

- Si le nom d'utilisateur correspond au propriétaire de *foo*, les autorisations du propriétaire sont testées;
- Sinon, si le groupe de *foo* correspond à l'un des membres de la liste des groupes, les autorisations de groupe sont testées;
- Sinon, les autres permissions de *foo* sont testées.

Si une vérification des autorisations échoue, l'opération client échoue.

1. Comprendre la mise en œuvre

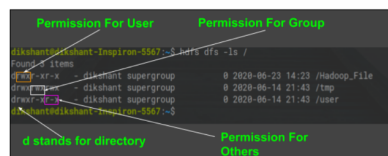
Chaque opération de fichier ou de répertoire transmet le nom de chemin complet au NameNode, et les vérifications des autorisations sont appliquées le long du chemin pour chaque opération. La structure client associera implicitement l'identité de l'utilisateur à la connexion au NameNode, réduisant ainsi le besoin de modifications de l'API client existante. Il a toujours été le cas que lorsqu'une opération sur un fichier réussit, l'opération peut échouer lorsqu'elle est répétée car le fichier, ou un répertoire sur le chemin, n'existe plus. Par exemple, lorsque le client commence à lire un fichier, il fait une première requête au NameNode pour découvrir l'emplacement des premiers blocs du fichier. Une deuxième demande faite pour trouver des blocs supplémentaires peut échouer. En revanche, la suppression d'un fichier ne révoque pas l'accès d'un client qui connaît déjà les blocs du fichier. Avec l'ajout d'autorisations, L'accès d'un client à un dossier peut être retiré entre les demandes. Là encore, la modification des autorisations ne révoque pas l'accès d'un client qui connaît déjà les blocs du fichier.

2. Comment pouvez-vous modifier l'autorisation de ce fichier HDFS ?

chmod qui signifie la commande de mode de changement est utilisé pour changer l'autorisation pour les fichiers dans notre HDFS.

Avec la commande **hadoop fs -ls /user/** on a les utilisateurs et leur droits sur les annuaires.

```
[root@bigdatalite ~]# hadoop fs -ls /user/
Found 10 items
drwxr-xr-x - bdd bdd 0 2017-12-22 10:51 /user/bdd
drwxrwxrwt - mapred hadoop 0 2017-12-21 17:57 /user/history
drwxrwx-- - hive hive 0 2017-12-21 17:31 /user/hive
drwxr-xr-x - hue hue 0 2017-12-21 17:33 /user/hue
drwxr-xr-x - impala impala 0 2017-12-21 17:31 /user/impala
drwxr-xr-x - oracle supergroup 0 2017-12-22 16:02 /user/odi
drwxrwxr-x - oozie oozie 0 2017-12-21 17:32 /user/oozie
drwxr-xr-x - oracle oracle 0 2018-01-03 16:16 /user/oracle
drwxr-xr-x - spark spark 0 2017-12-21 17:32 /user/spark
drwxrwxr-x - sqoop2 sqoop 0 2017-12-21 17:29 /user/sqoop2
[root@bigdatalite ~]#
```



The screenshot shows the command `hadoop fs -ls /user/hive` and its output. Annotations with arrows point to specific parts of the output:

- Permission For User:** Points to the permissions `drwxr-xr-x`.
- Permission For Group:** Points to the group `hive`.
- d stands for directory:** Points to the leading `d` in `drwxr-xr-x`.
- Permission For Others:** Points to the permissions `drwxrwx--`.

Lorsqu'on tape la commande **hadoop fs -ls /user/hive**, on voit qu'on n'a pas le droit de lister son contenu.

```
[root@bigdatalite ~]# hadoop fs -ls /user/hive
ls: Permission denied: user=root, access=READ_EXECUTE, inode="/user/hive":hive:hive:drwxrwx--
[root@bigdatalite ~]#
```

3. Nous allons donner les droits de lire sur le dossier HDFS a l'utilisateur Hadoop
- On se connecte en tant que **oracle**
 - On donne les priorites r ,w et x a l'utilisateur hadoop sur le dossier hive :**hadoop fs -chmod o+rx /user/hive**
 - En retappant la commande **hadoop fs -ls /user/hive** , on voit qu'on a pas le droit lister sont contenue.

```
[root@bigdatalite ~]# su oracle
[oracle@bigdatalite ~]# cd
[oracle@bigdatalite ~]$ hadoop fs -chmod o+rx /user/hdfs
chmod: '/user/hdfs': No such file or directory
[oracle@bigdatalite ~]$ hadoop fs -chmod o+rx /user/hive
[oracle@bigdatalite ~]$ su root
Password:
[root@bigdatalite oracle]# cd
[root@bigdatalite ~]# hadoop fs -ls /user/hive
Found 2 items
drwxr-xr-x  - hive hive          0 2017-12-21 17:31 /user/hive/sentry
drwxrwxrwx  - hive hive          0 2017-12-22 16:02 /user/hive/warehouse
[root@bigdatalite ~]#
```

Pour enlever les priorités on tape la commande **hadoop fs -chmod o-rwx /user/hive** en tant que user oracle.

2.5 Access Control liste (ACL)

En plus du modèle d'autorisations POSIX traditionnel, HDFS prend également en charge les ACL POSIX (listes de contrôle d'accès). Les ACL sont utiles pour implémenter des exigences d'autorisation qui diffèrent de la hiérarchie organisationnelle naturelle des utilisateurs et des groupes. Une ACL permet de définir différentes autorisations pour des utilisateurs nommés spécifiques ou des groupes nommés, pas seulement le propriétaire du fichier et le groupe du fichier. Par défaut, la prise en charge des ACL est désactivée et le NameNode interdit la création d'ACL. Pour activer la prise en charge des ACL, définissez `dfs.namenode.acls.enabled` sur `true` dans la configuration NameNode.

1. Activation des ACL HDFS à l'aide de Cloudera Manager

- Accédez à la console d'administration de Cloudera Manager et accédez au service HDFS . Cliquez sur l'onglet Configuration puis dans avancer .
- Sélectionnez Catégorie > custom hdfs site
- Cocher la proprieter **dfs.namenode.acls.enabled** .



- Entrez une raison de la modification , puis cliquez sur Enregistrer les modifications pour valider les modifications.

2. Manipulation ACL

Pour lister toutes les ACL situé dans `/user /hive` on tape la commande en tant que user hdfs : **hadoop fs -getfacl /user/hive** Pour donner à l'utilisateur root l'autorisation de lecture et

```
[oracle@bigdatalite ~]$ hadoop fs -getfacl /user/hive
# file: /user/hive
# owner: hive
# group: hive
user::rwx
group::rwx
other::rwx
[oracle@bigdatalite ~]$
```

d'écriture sur `/user/hive/` on tape la commande en tant que user oracle : **hadoop fs -setfacl -m user : root : r-x /user/hive**

L'image ci-dessous montre qu'on a appliqué une acl sur le dossier `/user/hive`

```
[oracle@bigdatalite ~]$ hadoop fs -ls /user
Found 10 items
drwxr-xr-x - bdd bdd 0 2017-12-22 10:51 /user/bdd
drwxrwxrwt - mapred hadoop 0 2017-12-21 17:57 /user/history
drwxrwxrwt+ - hive hive 0 2017-12-21 17:31 /user/hive
drwxr-xr-x - hue hue 0 2017-12-21 17:33 /user/hue
drwxr-xr-x - impala impala 0 2017-12-21 17:31 /user/impala
drwxr-xr-x - oracle supergroup 0 2017-12-22 16:02 /user/odi
drwxrwxr-x - oozie oozie 0 2017-12-21 17:32 /user/oozie
drwxr-xr-x - oracle oracle 0 2018-01-03 16:16 /user/oracle
drwxr-xr-x - spark spark 0 2017-12-21 17:32 /user/spark
drwxrwxr-x - sqoop2 sqoop 0 2017-12-21 17:29 /user/sqoop2
[oracle@bigdatalite ~]$
```

Pour enlever l'ACL on tape la commande **hadoop fs -setfacl -x user : root /user/hive** en tant que user oracle.

2.6 Authentification avec kerberos

1. Définition :

Kerberos est un moyen d'authentifier les utilisateurs qui a été développé au MIT et qui s'est développé pour devenir l'approche d'authentification la plus utilisée. Hadoop nécessite que Kerberos soit sécurisé car dans l'authentification par défaut, Hadoop et toutes les machines du cluster croient que toutes les informations d'identification des utilisateurs sont présentées. Pour surmonter cette vulnérabilité, kerberos fournit un moyen de vérifier l'identité des utilisateurs. La vérification d'identité Kerberos est implémentée via un modèle client / serveur. Plusieurs terminologies sont utilisées lors de l'implémentation de la vérification d'identité kerberos. Une identité qui doit être vérifiée est appelée principal. Les principaux sont divisés en deux catégories : les principaux d'utilisateur et les principaux de service.

2. Rôle du Kerberos :

- Contrôles d'accès au niveau des utilisateurs

Voici un mémoire sur les contrôles d'accès au niveau des utilisateurs :

- Les utilisateurs de Hadoop ne devraient être en mesure d'accéder qu'aux données autorisées pour eux
- Seuls les utilisateurs authentifiés devraient être en mesure de soumettre des emplois au cluster Hadoop
- Les utilisateurs doivent être en mesure de voir, de modifier et de tuer seulement leur propre emploi
- Seuls les services authentifiés devraient pouvoir s'inscrire DataNodes ou TaskTracker
- L'accès de bloc de données dans DataNode doit être sécurisé, et seulement les utilisateurs authentifiés doivent être en mesure d'accéder aux données stockées dans le Cluster Hadoop

- Contrôles d'accès au niveau du service

Voici une figure des contrôles d'accès au niveau du service :

- Authentification évolutive : les clusters Hadoop se composent d'un grand nombre de et les modèles d'authentification doivent être évolutifs pour grande authentification du réseau
- Usurpation d'identité : les services Hadoop devraient pouvoir se faire passer pour l'utilisateur soumettre le travail afin que l'isolement correct de l'utilisateur puisse être maintenu
- Self-Served : Hadoop emplois fonctionnent pendant de longues durées, de sorte qu'ils devraient être en mesure de s'assurer que les emplois sont en mesure d'auto-renouveler l'authentification déléguée de l'utilisateur pour terminer le travail
- Secure IPC : Les services Hadoop devraient être en mesure de s'authentifier les uns les autres et assurer une communication sécurisée entre eux

3. Implémentation de Kerberos

Pour implémenter l'authentification Kerberos dans Hadoop, plusieurs étapes sont requises et elles sont répertoriées ci-dessous. `ssh root@sandbox-hdp.hortonworks.com -p 2222`

- (a) Clonez notre serveur Ambari de référentiel github dans votre cluster HDP Remarque - Ce script installera et configurera KDC sur votre serveur Ambari.

```
[root@sandbox-hdp ~]# git clone https://github.com/crazyadmins/useful-scripts.git
Cloning into 'useful-scripts'...
remote: Enumerating objects: 476, done.
remote: Total 476 (delta 0), reused 0 (delta 0), pack-reused 476
Receiving objects: 100% (476/476), 76.20 KiB | 0 bytes/s, done.
Resolving deltas: 100% (238/238), done.
```

- (b) Accédez au répertoire utile-scripts /ambari

```
[root@sandbox-hdp ambari]# ls -lrt
total 44
-rw-r--r-- 1 root root  987 Oct 20 13:21 setup_only_kdc.sh
-rw-r--r-- 1 root root 6368 Oct 20 13:21 setup_kerberos.sh
-rwxr-xr-x 1 root root 1099 Oct 20 13:21 restore_db.sh
-rw-r--r-- 1 root root  914 Oct 20 13:21 README-SETUP-KERBEROS
-rw-r--r-- 1 root root  747 Oct 20 13:21 README-SETUP-KDC-ONLY
-rw-r--r-- 1 root root  417 Oct 20 13:21 krb5.conf.default
-rw-r--r-- 1 root root  366 Oct 20 13:21 ambari.props
-rwxr-xr-x 1 root root 10390 Oct 20 13:21 ambari-admin.sh
```

- (c) Copiez setupkerberos.sh et ambari.props sur l'hôte sur lequel vous souhaitez configurer le serveur KDC

- (d) Editez et modifiez le fichier ambari.props en fonction de votre environnement de cluster

```
[root@sandbox-hdp ambari]# cat ambari.props
CLUSTER_NAME=Sandbox
AMBARI_ADMIN_USER=admin
AMBARI_ADMIN_PASSWORD=Mourtalla
AMBARI_HOST=sandbox-hdp.hortonworks.com
KDC_HOST=sandbox-hdp.hortonworks.com
REALM=HWX.COM
KERBEROS_CLIENTS=sandbox-hdp.hortonworks.com

##### Notes #####
#1. KERBEROS_CLIENTS - Comma separated list of Kerberos clients in case of multinode cluster
#2. Admin principal is admin/admin and password is hadoop
[root@sandbox-hdp ambari]#
```

- (e) Démarrez l'installation en exécutant simplement setupkerberos.sh

```
[root@sandbox-hdp ambari]# sh setup_kerberos.sh
```

Eplication du Script

- i. Pour configurer Kerberos, nous devons installer le centre de distribution de clés (KDC) :
yum install krb5-server krb5-libs krb5-workstation
- ii. Nous examinerons la version 5 du MIT Kerberos. Cela a trois fichiers de configuration. Le fichier **krb5.conf** est conservé dans le dossier / etc /. Les fichiers **kdc.conf** et **kadm5.acl** sont placés dans le dossier / usr / local / var / krb5kdc. krb5.conf est une configuration de niveau supérieur et fournit la configuration relative à l'emplacement des KDC, des serveurs d'administration et des mappages de noms d'hôte avec Kerberos royaumes.

```
[root@sandbox-hdp ~]# cat /etc/krb5.conf
[libdefaults]
    renew_lifetime = 7d
    forwardable= true
    default_realm = HWX.COM
    ticket_lifetime = 24h
    dns_lookup_realm = false
    dns_lookup_kdc = false
    #default_tgs_etypes = aes des3-cbc-sha1 rc4 des-cbc-md5
    #default_tkt_etypes =aes des3-cbc-sha1 rc4 des-cbc-md5

[logging]
    default = FILE:/var/log/krb5kdc.log
    admin_server = FILE:/var/log/kadmind.log
    kdc = FILE:/var/log/krb5kdc.log

[realms]
    HWX.COM = {
        admin_server = sandbox-hdp.hortonworks.com
        kdc = sandbox-hdp.hortonworks.com
    }

[root@sandbox-hdp ~]# █
```

```
[root@sandbox-hdp ~]# cat /var/kerberos/krb5kdc/kadm5.acl
*/admin@HWX.COM *
[root@sandbox-hdp ~]# cat /var/kerberos/krb5kdc/kdc.conf
[kdcdefaults]
    kdc_ports = 88
    kdc_tcp_ports = 88

[realms]
    EXAMPLE.COM = {
        #master_key_type = aes256-cts
        acl_file = /var/kerberos/krb5kdc/kadm5.acl
        dict_file = /usr/share/dict/words
        admin_keytab = /var/kerberos/krb5kdc/kadm5.keytab
        supported_etypes = aes256-cts:normal aes128-cts:normal des3-hmac-sha1:normal arcfour
        mellia128-cts:normal des-hmac-sha1:normal des-cbc-md5:normal des-cbc-crc:normal
    }
[root@sandbox-hdp ~]#
```

- iii. L'étape suivante consiste à créer une base de données KDC pour notre installation. Exécutez la commande suivante :**kdb5_util create -r HWX.COM -s**
- iv. Une fois la base de données KDC créée, le principal administrateur doit être configuré dans la base de données. Pour ce faire, ajoutez d'abord le principal administrateur dans le fichier / var / Fichier Kerberos / krb5kdc / kadm.acl contenant la liste de contrôle d'accès (ACL) est utilisé par le démon kadmind pour gérer l'accès à la base de données Kerberos. Un fichier kadm.acl typique qui fournit à tous les administrateurs un privilège complet aura l'entrée suivante :***/admin@HWX.COM**
- v. Une fois les configurations correctement définies, nous sommes prêts à démarrer les démons Kerberos :**service kadmin start**

- vi. Ensuite, nous configurons le mot de passe principal dans la base de données KDC à l'aide de `kadmin`. commande locale sur le serveur KDC maître. Exécutez la commande suivante pour configurer le principal administrateur et indiquez le mot de passe du principal administrateur : **`kadmin.local -p admin/admin`**
- vii. Une fois la configuration de l'utilisateur administrateur terminée et les démons Kerberos démarrés, puis nous pouvons ajouter les principaux à la base de données Kerberos à l'aide de l'utilitaire `kadmin` : **`add_principal -randkey admin/hortonworks.com@HWX.COM`**

```
echo -e "\n`ts` Installing kerberos RPMs"
yum -y install krb5-server krb5-libs krb5-workstation
echo -e "\n`ts` Configuring Kerberos"
sed -i.bak "s/EXAMPLE.COM/$REALM/g" $LOC/krb5.conf.default
sed -i.bak "s/kerberos.example.com/$KDC_HOST/g" $LOC/krb5.conf.default
cat $LOC/krb5.conf.default > /etc/krb5.conf
kdb5_util create -s -P hadoop
echo -e "\n`ts` Starting KDC services"
service krb5kdc start
service kadmin start
chkconfig krb5kdc on
chkconfig kadmin on
echo -e "\n`ts` Creating admin principal"
kadmin.local -q "addprinc -pw hadoop admin/admin"
sed -i.bak "s/EXAMPLE.COM/$REALM/g" /var/kerberos/krb5kdc/kadm5.acl
echo -e "\n`ts` Restarting kadmin"
service kadmin restart
```

viii. Configurer Hadoop avec Kerberos authentication :

Une fois la configuration Kerberos terminée et les principaux utilisateurs ajoutés au KDC, nous pouvons configurer Hadoop pour utiliser l'authentification Kerberos. Nous commencerons la configuration utilisant Cloudera Distribution de Hadoop.

- ix. Dans HDP, trois utilisateurs (`hdfs`, `mapred` et `yarn`) sont utilisés pour exécuter les divers démons Hadoop. Tous les systèmes de fichiers distribués Hadoop (HDFS) les démons tels que `NameNode`, `DataNode` et `NameNode` secondaire sont exécutés sous l'utilisateur `hdfs`, tandis que pour `MRV1`, les démons liés à MapReduce tels que `JobTracker` et `TaskTracker` s'exécutent à l'aide de l'utilisateur `mapred`. Nous devons créer les principaux `hdfs`, `mapred` et `yarn` dans KDC pour garantir Kerberos authentification pour les démons Hadoop. Nous avons des services `http` exposés par tous ces services, nous devons donc également créer un principal de service `http`. Nous utilisons ce qui suit commandes `kadmin` pour créer ces principaux : **`kadmin`**

`kadmin : addprinc -randkey hdfs/hortonworks.com@HWX.COM`

`kadmin : addprinc -randkey mapred/hortonworks.com@HWX.COM`

`kadmin : addprinc -randkey http/hortonworks.com@HWX.COM`

`kadmin : addprinc -randkey yarn/hortonworks.com@HWX.COM`

- x. Un keytab est un fichier contenant des paires de principaux Kerberos et des clés chiffrées dérivées à partir du mot de passe Kerberos. Ce fichier est utilisé pour l'authentification sans tête avec KDC lorsque les services fonctionnent en arrière-plan sans intervention humaine. Le keytab Le fichier est créé à l'aide des commandes `kadmin`.

Les utilisateurs `hdfs` et `mapred` exécutent plusieurs démons Hadoop en arrière-plan, nous besoin de créer le fichier keytab pour les utilisateurs `hdfs` et `mapred`. Nous devons également ajouter le principal `http` de ces keytabs, de sorte que l'interface utilisateur Web associée à Hadoop soit authentifié à l'aide de Kerberos.

`kadmin : xst -norandkey -k hdfs.keytab hdfs/hortonworks.com@HWX.COM`

http/ hortonworks.com@HWX.COM

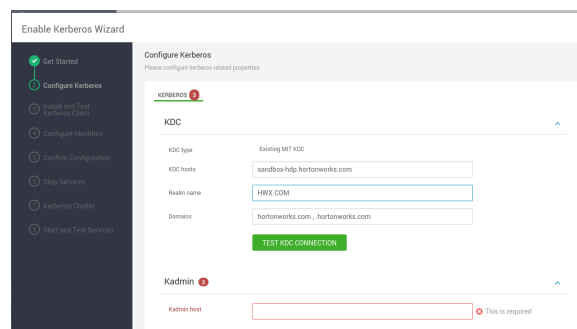
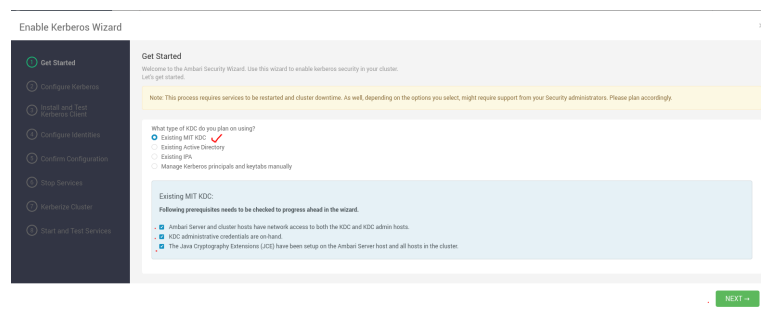
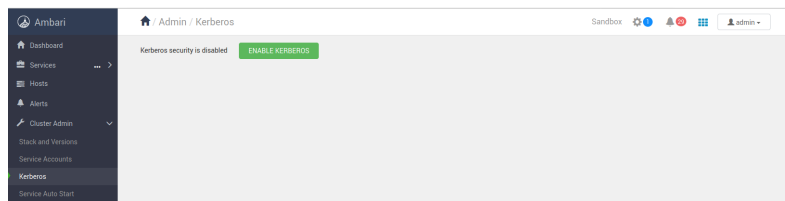
kadmin : xst -norandkey -k mapred.keytab hdfs/hortonworks.com@HWX.COM

http/hortonworks.com@HWX.COM

kadmin : xst -norandkey -k yarn.keytab hdfs/hortonworks.com@HWX.COM

http/ hortonworks.com@HWX.COM

- xi. Lorsque vous choisissez KDC MIT existant , l'assistant Kerberos vous demande des informations relatives au KDC, au compte administrateur KDC et aux principaux Service et Ambari. Une fois fournis, Ambari créera automatiquement des principaux, générera des keytabs et distribuera des keytabs aux hôtes du cluster. Les services seront configurés pour Kerberos et les composants de service sont redémarrés pour s'authentifier auprès du KDC.



(f) Ambari apres le lancement du scripte

Ambari / Admin / Kerberos

Sandbox

Kerberos security is enabled

DISABLE KERBEROS REGENERATE KEYTABS DOWNLOAD CSV

GENERAL ADVANCED

Global

Keytab Dir	/etc/security/keytabs
Realm	\${kerberos-env/realm}
Additional Realms	
Principal Suffix	-\${cluster_name toLower()}
Spnego Keytab	\${keytab_dir}/spnego.service.keytab
Spnego Principal	HTTP/_HOST@\${realm}

Ambari Principals

Smoke user keytab	\${keytab_dir}/smokeuser.headless.keytab
Smoke user principal	\${cluster-env/smokeuser}\${principal_suffix}@\${realm}

4. Manipulation avec le fichier HDFS

On vas essayer de lister le contenu du dossier /user

```
[root@sandbox-hdp ~]# hadoop fs -ls /user
20/10/23 20:39:16 WARN ipc.Client: Exception encountered while connecting to the server : org.apache.hadoop.security.AccessCon
trolException: Client cannot authenticate via:[TOKEN, KERBEROS]
ls: DestHost:destPort sandbox-hdp.hortonworks.com:8020 , LocalHost:localPort sandbox-hdp.hortonworks.com/172.18.0.2:0. Failed
on local exception: java.io.IOException: org.apache.hadoop.security.AccessControlException: Client cannot authenticate via:[TO
KEN, KERBEROS]
[root@sandbox-hdp ~]#
```

On voit que l'accès est refusé car on a pas été authentifier par Kerberos.

Nous allons nous authentifier en tant qu'utilisateur administratif dans le domaine HWX grace a la commande :

```
[root@sandbox-hdp ~]# kinit admin/admin@HWX.COM
Password for admin/admin@HWX.COM:
```

voir quelles informations d'identification Kerberos, le cas échéant, ils ont le cache de leurs informations d'identification . Le cache des informations d'identification est l'endroit sur le système de fichiers local où, lors de l'authentification réussie auprès de l'AS, les TGT sont stockés. Par défaut, cet emplacement est généralement le fichier / tmp / krb5cc uid où uid est l'ID utilisateur numérique sur le système local. Après un succès kinit

```
[root@sandbox-hdp ~]# klist
Ticket cache: FILE:/tmp/krb5cc_0
Default principal: admin/admin@HWX.COM

Valid starting    Expires          Service principal
10/23/2020 20:20:28  10/24/2020 20:20:28  krbtgt/HWX.COM@HWX.COM
```


On va essayer de lister le contenu du dossier /user

```
[root@sandbox-hdp ~]# hadoop fs -ls /user
Found 15 items
drwxr-xr-x - admin hdfs 0 2018-11-29 17:28 /user/admin
drwxrwx--- - ambari-qa hdfs 0 2018-11-29 17:25 /user/ambari-qa
drwxr-xr-x - amy_ds hdfs 0 2018-11-29 17:28 /user/amy_ds
drwxr-xr-x - anonymous hdfs 0 2018-11-29 17:28 /user/anonymous
drwxr-xr-x - druid hadoop 0 2018-11-29 19:01 /user/druid
drwxr-xr-x - hbase hdfs 0 2018-11-29 17:48 /user/hbase
drwx----- - hdfs hdfs 0 2018-11-29 19:21 /user/hdfs
drwxr-xr-x - hive hdfs 0 2018-11-29 19:04 /user/hive
drwxrwxr-x - livy hdfs 0 2018-11-29 17:55 /user/livy
drwxr-xr-x - maria_dev hdfs 0 2018-11-29 17:28 /user/maria_dev
drwxrwxr-x - oozie hdfs 0 2018-11-29 19:06 /user/oozie
drwxr-xr-x - raj_ops hdfs 0 2018-11-29 17:28 /user/raj_ops
drwxr-xr-x - root hdfs 0 2018-11-29 17:28 /user/root
drwxrwxr-x - spark hdfs 0 2018-11-29 17:57 /user/spark
drwxr-xr-x - zeppelin hdfs 0 2018-11-29 17:50 /user/zeppelin
[root@sandbox-hdp ~]# █
```

Puisque nous nous sommes identifiés on peut donc manipuler

Pour détruire les informations d'identification dans le cache d'informations d'identification.

```
[root@sandbox-hdp ~]# kdestroy
[root@sandbox-hdp ~]# klist
klist: No credentials cache found (filename: /tmp/krb5cc_0)
[root@sandbox-hdp ~]#
```

2.7 Conclusion

Pour augmenter la sécurité autour du Big Data, plusieurs solutions sont à la disposition des entreprises. Il est possible de collaborer avec d'autres entreprises de l'industrie pour créer des standards et partager les meilleures pratiques à adopter. Il faut déployer un système de chiffrement efficace pour protéger les données personnelles partagées par des tiers. Les logiciels open source comme Hadoop réclament une attention particulière.

On a vu que hadoop permet de manipuler les big data et il était donc plus que nécessaire de le sécuriser. De manière native Hadoop n'est pas très sécurisé; c'est là qu'intervient Kerberos qui va permettre aux utilisateurs de s'identifier pour exécuter des tâches.

Cependant, il n'existe pas encore de plateforme permettant de prédire avec exactitude quand une faille de sécurité risque d'apparaître. Les entreprises doivent utiliser les processus de Machine Learning et Big Data pour détecter les attaques de façon précoce. L'objectif est d'empêcher les dégâts.

2.8 Webographie

Telecharger Oracle Big Data Lite : <https://www.oracle.com/database/technologies/bigdatalite-v411.html>

Telecharger VirtualBox : <https://www.virtualbox.org/>

Telecharger Docker : <https://www.docker.com/>

Telecharger HDP : <https://www.cloudera.com/downloads/hortonworks-sandbox/hdp.html>

Kerberos : [gitclonehttps://github.com/crazyadmins/useful-scripts.git](https://github.com/crazyadmins/useful-scripts.git)